10/622,572

日　本　国　特　許　庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出　願　年　月　日
Date of Application:　２００３年　７月１６日

出　願　番　号
Application Number:　特願２００３－１９７８５０

[ST. 10/C]:　[Ｊ Ｐ ２００３－１９７８５０]

出　願　人
Applicant(s):　株式会社リコー

２００３年　８月　７日

特許庁長官
Commissioner,
Japan Patent Office　今井康夫

出証番号　出証特２００３－３０６３５１３

【書類名】　特許願
【整理番号】　0304559
【提出日】　平成15年 7月16日
【あて先】　特許庁長官　今井　康夫　殿
【国際特許分類】　G03G 21/00
【発明の名称】　情報処理装置及び情報処理方法
【請求項の数】　15
【発明者】
　【住所又は居所】　東京都大田区中馬込１丁目３番６号　株式会社リコー内
　【氏名】　高橋　征司
【特許出願人】
　【識別番号】　000006747
　【氏名又は名称】　株式会社リコー
【代理人】
　【識別番号】　100070150
　【弁理士】
　【氏名又は名称】　伊東　忠彦
【先の出願に基づく優先権主張】
　【出願番号】　特願2002-212300
　【出願日】　平成14年 7月22日
【手数料の表示】
　【予納台帳番号】　002989
　【納付金額】　21,000円
【提出物件の目録】
　【物件名】　明細書　1
　【物件名】　図面　1
　【物件名】　要約書　1
　【包括委任状番号】　9911477
【プルーフの要否】　要

【書類名】　明細書

【発明の名称】　情報処理装置及び情報処理方法

【特許請求の範囲】

【請求項１】　ネットワークを介して接続される端末から送信されるＷｅｂ情報を要求する第一要求に基づいて該端末の端末種別を特定し、該特定した端末種別を示す端末種別情報を該第一要求に対する該Ｗｅｂ情報へのパスに付加することにより作成した基準パスを含み、該端末から自動的に該基準パスへアクセスさせる基準Ｗｅｂ情報を生成する基準Ｗｅｂ情報生成手段と、

前記第一要求に対する応答として前記基準Ｗｅｂ情報を前記端末へ送信し、前記基準パスによって特定される前記Ｗｅｂ情報を要求する第二要求を受信する通信手段とを有することを特徴とする情報処理装置。

【請求項２】　前記通信手段は、前記ネットワークから前記第一要求を受信した場合、該第一要求に対する前記Ｗｅｂ情報へのパスにおいて、前記端末種別のデフォルト値と前記基準Ｗｅｂ情報生成手段を識別する基準Ｗｅｂ情報識別子とを付加し、

前記基準Ｗｅｂ情報生成手段は、前記通信手段によって付加された前記基準Ｗｅｂ情報識別子によって実行され、前記特定した端末種別で前記デフォルト値を置き換えることを特徴とする請求項１記載の情報処理装置。

【請求項３】　前記通信手段は、前記第一要求に対する前記Ｗｅｂ情報へのパスにおいて、該Ｗｅｂ情報を識別する前記Ｗｅｂ情報識別子より前に前記デフォルト値を付加して前記基準パスを作成することを特徴とする請求項１又は２記載の情報処理装置。

【請求項４】　前記Ｗｅｂ情報を生成するＷｅｂ情報生成手段と、

前記第二要求によって指定される前記基準パスから取得した前記端末種別情報に基づいて、前記端末に応じた前記Ｗｅｂ情報を該端末に表示するための表示形式で記述することによって、Ｗｅｂページを生成する表示情報生成手段とを有することを特徴とする請求項１乃至３のいずれか一項記載の情報処理装置。

【請求項５】　前記表示情報生成手段は、

前記Ｗｅｂ情報生成手段によって生成された前記Ｗｅｂ情報と、前記端末種別

情報とをＸＭＬで記述するＸＭＬ記述手段と、

前記端末種別情報に基づいて、前記ＸＭＬによって記述された前記Ｗｅｂ情報を該Ｗｅｂ情報に応じたスタイルシートに従ってＨＴＭＬに変換することによって、前記Ｗｅｂページを生成するＨＴＭＬ変換手段とを有することを特徴とする請求項１乃至４のいずれか一項記載の情報処理装置。

【請求項６】　前記Ｗｅｂ情報から相対パスによってリンクされる該Ｗｅｂ情報とは異なる他のＷｅｂ情報を生成する複数の他のＷｅｂ情報生成手段を有し、

前記Ｗｅｂ情報が表示される前記端末にてユーザによって選択されることによって要求される前記他のＷｅｂ情報の第三要求に応じて、該他のＷｅｂ情報に対応する前記他のＷｅｂ情報生成手段が該他のＷｅｂ情報を生成すると、前記表示情報生成手段は、前記基準パスに設定されている前記端末種別情報に基づいて、前記端末に応じた前記他のＷｅｂ情報を該端末に表示するＷｅｂページを生成することを特徴とする請求項１乃至５のいずれか一項記載の情報処理装置。

【請求項７】　複数のフレーム毎に表示すべき前記Ｗｅｂ情報及び前記他のＷｅｂ情報の相対パスを設定し、Ｗｅｂページを分割する該複数のフレームを定義したＷｅｂフレーム情報を生成するＷｅｂフレーム情報生成手段を有し、

前記基準Ｗｅｂ情報生成手段は、前記端末種別情報を前記Ｗｅｂフレーム情報へのパスに付加することにより作成した前記基準パスを含み、該端末から自動的に該パスへアクセスさせる前記基準Ｗｅｂ情報を生成し、

前記通信手段が、前記基準Ｗｅｂ情報を前記端末の前記第一要求に対する応答として送信し、前記基準パスによって該端末から前記Ｗｅｂフレーム情報を要求する前記第二要求を受信すると、前記Ｗｅｂフレーム情報生成手段を実行することを特徴とする請求項６記載の情報処理装置。

【請求項８】　前記表示情報生成手段は、

前記第二要求によって指定される前記基準パスから取得した前記端末種別情報に基づいて前記Ｗｅｂフレーム情報生成手段を無効とし、前記第一要求によって要求された前記Ｗｅｂ情報の相対パスによって前記端末から該Ｗｅｂ情報を直接アクセスする前記Ｗｅｂページを生成することを特徴とする請求項７記載の情報

処理装置。

【請求項９】　画像を形成する画像形成手段と、

前記画像形成手段を制御する画像形成制御手段とを有し、

前記Ｗｅｂ情報生成手段と前記他のＷｅｂ情報生成手段との少なくとも１つのＷｅｂ情報生成手段は、前記画像形成制御手段から前記画像形成手段に関する情報を取得して、その取得した情報に基づいて前記Ｗｅｂ情報を生成することを特徴とする請求項６乃至８のいずれか一項記載の情報処理装置。

【請求項１０】　前記表示情報生成手段は、

前記基準パスに基づいて共通に設定される前記端末種別情報に基づいて、前記Ｗｅｂ情報生成手段又は前記他のＷｅｂ情報生成手段によって生成された前記Ｗｅｂ情報又は前記他のＷｅｂ情報にイメージを付加した前記Ｗｅｂページを生成することを特徴とする請求項５乃至８のいずれか一項記載の情報処理装置。

【請求項１１】　前記表示情報生成手段は、

前記基準パスに基づいて共通に設定される前記端末種別情報に基づいて、前記Ｗｅｂ情報生成手段又は前記他のＷｅｂ情報生成手段によって生成された前記Ｗｅｂ情報又は前記他のＷｅｂ情報を前記端末に対応した文字サイズで表示する前記Ｗｅｂページを生成することを特徴とする請求項６乃至１０のいずれか一項記載の情報処理装置。

【請求項１２】　前記表示情報生成手段は、

前記基準パスに基づいて共通に設定される前記端末種別情報に基づいて、前記Ｗｅｂ情報生成手段又は前記他のＷｅｂ情報生成手段によって生成された前記Ｗｅｂ情報又は前記他のＷｅｂ情報を前記端末に対応した文字数で表示する前記Ｗｅｂページを生成することを特徴とする請求項６乃至１１のいずれか一項記載の情報処理装置。

【請求項１３】　前記基準Ｗｅｂ情報生成手段と、前記Ｗｅｂ情報生成手段と、前記他のＷｅｂ情報生成手段と、前記Ｗｅｂフレーム情報生成手段とは、Ｃ言語によって開発されたプログラムであることを特徴とする請求項７乃至１２のいずれか一項記載の情報処理装置。

【請求項１４】　ネットワークを介して接続される端末から送信されるＷｅ

ｂ情報を要求する第一要求に基づいて該端末の端末種別を特定し、該特定した端末種別を示す端末種別情報を該第一要求に対する該Ｗｅｂ情報へのパスに付加することにより作成した基準パスを含み、該端末から自動的に該基準パスへアクセスさせる基準Ｗｅｂ情報を生成する基準Ｗｅｂ情報生成手順と、

前記第一要求に対する応答として送信する基準Ｗｅｂ情報を前記端末へ送信し、前記基準パスによって特定される前記Ｗｅｂ情報を要求する第二要求を受信する通信手順とを有することを特徴とする情報処理方法。

【請求項１５】　前記Ｗｅｂ情報を生成するＷｅｂ情報生成手順と、

前記第二要求によって指定される前記基準パスから取得した前記端末種別情報に基づいて、前記端末に応じた前記Ｗｅｂ情報を該端末に表示するための表示形式で記述することによって、Ｗｅｂページを生成する表示情報生成手順とを有することを特徴とする請求項１４記載の情報処理方法。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】

本発明は、Ｗｅｂアプリケーションを跨るページ遷移の際にもアクセスしている端末種別の継承を可能とし、ページ遷移の際のユーザの利便性を損なうことなく端末の画面表示域に応じた適切な情報提供を可能とする、複数のＷｅｂアプリケーションを有する情報処理装置を提供するものである。

【０００２】

【従来の技術】

近年、インターネットを利用した情報提供が、クライアントＰＣ（Personal Computer）のみならず、携帯電話器及びＰＤＡ（Personal Digital Assistants）端末等の移動体端末へ拡大し、また、その情報提供の方法が多様化してきている。

【０００３】

クライアントＰＣ、携帯電話器及びＰＤＡ端末等の端末側での表示画面のサイズに応じた情報提供をするために、情報提供用に予め用意してあるＨＴＭＬ(HyperText Markup Language)データに対して、端末の種別に応じた表示内容を生成

することが提案されている（例えば、特許文献１参照。）。或いは、情報提供を行うＷｅｂサーバと情報要求を行う端末とを仲介するゲートウェイサーバが、端末からの情報要求に対するＷｅｂサーバから提供される情報をその端末の種別に応じて不要なデータ除去や画像の色属性の調整作業といった付加的作業を行うことが提案されている（例えば、特許文献２参照。）。

【０００４】

一方、近年、画像形成に関する情報を提供可能とするＷｅｂサーバ機能を有するデータ処理装置が提案されている（例えば、特許文献３参照。）。このデータ処理装置によって、ユーザはインターネットを介すのみでデータ処理装置に備えられた画像形成処理を行う機器に関する情報の提供を受けることができる。

【０００５】

また、複数のページで構成された情報提供において、クライアントＰＣ側にてページ遷移が行われた場合でも、クライアントＰＣを利用しているユーザが入力した情報を所定領域に格納し、以後ページが遷移した際にその入力情報を用いることによって継承させて情報を提供するデータ継承方式が提案されている（例えば、特許文献４参照。）。

【０００６】

【特許文献１】

特開平１１－１７５５１５号公報

【０００７】

【特許文献２】

特開２００２－２３１０８号公報

【０００８】

【特許文献３】

特開２００２－７０９５号公報

【０００９】

【特許文献４】

特開平１１－３０６０７０号公報

【００１０】

【発明が解決しようとする課題】

しかしながら、上記従来のような情報提供の方法には、以下のような問題があった。

【００１１】

画像形成に関する情報を提供可能とするＷｅｂサーバ機能を有するＷｅｂサーバにおいて、画像形成処理又はその画像形成処理を行うＷｅｂサーバによって制御されるプロッタ等の機器の状態を情報として提供するような場合、その時点での状態であるため、予めＨＴＭＬデータを用意しておくことができない。したがって、特許文献１で提案されている方法を適用することができない。また、特許文献２では、ゲートウェイサーバを備えるための費用及びメンテナンス等がかかってしまう。

【００１２】

特許文献３において提案されるデータ処理装置では、クライアントＰＣ、携帯電話器及びＰＤＡ端末等の端末側での表示画面のサイズに応じた情報提供を行うことができない。

【００１３】

更に、特許文献４において提案されるデータ継承方式では、ユーザに関する情報を入力させるためのページが必要であったり、ユーザが明示的にそのページへ入力する必要がある。また、ネットワーク接続されている間ユーザ毎に継承させるべき情報を管理しなければならない。従って、表示画面のサイズに応じた情報提供を自動的に行うことができない。

【００１４】

そこで、本発明の課題は、Ｗｅｂアプリケーションを跨るページ遷移の際にもアクセスしている端末種別の継承を可能とし、ページ遷移の際のユーザの利便性を損なうことなく端末の画面表示域に応じた適切な情報提供を可能とする、複数のＷｅｂアプリケーションを有する情報処理装置を提供することである。

【００１５】

【課題を解決するための手段】

上記課題を解決するため、本発明は、請求項１に記載されるように、ネットワ

ークを介して接続される端末から送信されるＷｅｂ情報を要求する第一要求に基づいて該端末の端末種別を特定し、該特定した端末種別を示す端末種別情報を該第一要求に対する該Ｗｅｂ情報へのパスに付加することにより作成した基準パスを含み、該端末から自動的に該基準パスへアクセスさせる基準Ｗｅｂ情報を生成する基準Ｗｅｂ情報生成手段と、前記第一要求に対する応答として送信する基準Ｗｅｂ情報を前記端末へ送信し、前記基準パスによって特定される前記Ｗｅｂ情報を要求する第二要求を受信する通信手段とを有するように構成される。

【００１６】

このような情報処理装置では、端末から基準パスで自動的にアクセスさせるＷｅｂ情報を生成して提供するため、端末側では、ユーザによる端末種別情報の入力を行なうことなく、このＷｅｂ情報によって端末種別情報を含む基準パスを取得することができる。また、ユーザが求めるＷｅｂ情報を提供する前に基準パスが提供されるため、Ｗｅｂ情報を生成するＷｅｂアプリケーションの開発者は、端末種別毎の開発を行わなくてよい。

【００１７】

前記Ｗｅｂ情報は、例えば、インターネットを介してＷｅｂブラウザ上で提供される情報である。

【００１８】

更に、本発明は、請求項２に記載されるように、前記通信手段は、前記ネットワークから前記第一要求を受信した場合、該第一要求に対する該Ｗｅｂ情報へのパスにおいて、前記端末種別のデフォルト値と前記基準Ｗｅｂ情報生成手段を識別する基準Ｗｅｂ情報識別子とを付加し、前記基準Ｗｅｂ情報生成手段は、前記通信手段によって付加された前記基準Ｗｅｂ情報識別子によって前記基準Ｗｅｂ情報生成手段が実行され、該特定した端末種別で前記デフォルト値を置き換えるように構成することができる。

【００１９】

このような情報処理装置では、ネットワークから第一要求を受信すると、通信手段（例えば、ＨＴＴＰデーモン）が予めデフォルト値と基準Ｗｅｂ情報識別子とを付加するため、第一要求に対して常に基準Ｗｅｂ情報生成手段を実行させる

ことができる。

【００２０】

また、本発明は、請求項３に記載されるように、前記通信手段は、前記第一要求に対する前記Ｗｅｂ情報へのパスにおいて、該Ｗｅｂ情報を識別する前記Ｗｅｂ情報識別子より前に前記デフォルト値を付加して前記基準パスを作成するように構成することができる。

【００２１】

このような情報処理装置では、端末種別情報がＷｅｂ情報識別子より前に設定されるため、Ｗｅｂ情報識別子以降を相対パスとして設定されるようにすることができる。従って、基準Ｗｅｂ情報からページ遷移した場合において、常に端末種別情報が継承される。

【００２２】

更に、本発明は、請求項４に記載されるように、前記Ｗｅｂ情報を生成するＷｅｂ情報生成手段と、前記第二要求によって指定される前記基準パスから取得した前記端末種別情報に基づいて、前記端末に応じた前記Ｗｅｂ情報を該端末に表示するための表示形式で記述することによって、Ｗｅｂページを生成する表示情報生成手段とを有するように構成することができる。また、本発明は、請求項５に記載されるように、前記表示情報生成手段は、前記Ｗｅｂ情報生成手段によって生成された前記Ｗｅｂ情報と、前記端末種別情報とをＸＭＬで記述するＸＭＬ記述手段と、前記端末種別情報に基づいて、前記ＸＭＬによって記述された前記Ｗｅｂ情報を該Ｗｅｂ情報に応じたスタイルシートに従ってＨＴＭＬに変換することによって、前記Ｗｅｂページを生成するＨＴＭＬ変換手段とを有するように構成できる。

【００２３】

このような情報処理装置では、表示情報生成手段によって端末種別に対応したＷｅｂページの生成が行われ、また、Ｗｅｂ情報の生成に依存しないため、Ｗｅｂアプリケーションの追加を容易に実現することができる。

【００２４】

また、本発明は、請求項６に記載されるように、前記Ｗｅｂ情報から相対パス

によってリンクされる該Ｗｅｂ情報とは異なる他のＷｅｂ情報を生成する複数の他のＷｅｂ情報生成手段を有し、前記Ｗｅｂ情報が表示される前記端末にてユーザによって選択されることによって要求される前記他のＷｅｂ情報の第三要求に応じて、該他のＷｅｂ情報に対応する前記他のＷｅｂ情報生成手段が該他のＷｅｂ情報を生成すると、前記表示情報生成手段は、前記基準パスに設定される前記端末種別情報に基づいて、前記端末に応じた前記他のＷｅｂ情報を該端末に表示するＷｅｂページを生成するように構成することができる。

【００２５】

このような情報処理装置において、端末へ提供されたＷｅｂ情報から他のＷｅｂ情報が相対パスによってリンクされた場合においても、基準パスに基づいて共通に設定される端末種別情報を常に取得することができるため、常に、端末の表示画面に応じたＷｅｂページ生成することができる。

【００２６】

更に、本発明は、請求項７に記載されるように、複数のフレーム毎に表示すべき前記Ｗｅｂ情報及び前記他のＷｅｂ情報の相対パスを設定し、Ｗｅｂページを分割する該複数のフレームを定義したＷｅｂフレーム情報を生成するＷｅｂフレーム情報生成手段を有し、前記基準Ｗｅｂ情報生成手段は、前記端末種別情報を前記Ｗｅｂフレーム情報へのパスに付加することにより作成した前記基準パスを含み、該端末から自動的に該パスへアクセスさせる前記基準Ｗｅｂ情報を生成し、前記通信手段が、前記基準Ｗｅｂ情報を前記端末の前記第一要求に対する応答として送信し、前記基準パスによって該端末から前記Ｗｅｂフレーム情報を要求する前記第二要求を受信すると、前記Ｗｅｂフレーム情報生成手段を実行するように構成することができる。

【００２７】

このような情報処理装置において、端末にフレームによって分割されたＷｅｂページを提供する場合においても、基準パスを設定したＷｅｂフレーム情報を端末側から要求するようにするため、各フレームからアクセスされるＷｅｂ情報及び他のＷｅｂ情報は相対パスによってアクセスすることができ、且つ、端末種別情報を継承させることができる。

【００２８】

また、本発明は、請求項８に記載されるように、前記表示情報生成手段は、前記第二要求によって指定される前記基準パスから取得した前記端末種別情報に基づいて、前記Ｗｅｂフレーム情報生成手段を無効とし、前記第一要求によって要求された前記Ｗｅｂ情報の相対パスによって前記端末側から該Ｗｅｂ情報が直接アクセスする前記Ｗｅｂページを生成するように構成することができる。

【００２９】

このような情報処理装置において、端末がＰＣ以外の場合にはフレーム情報を端末に送信せず直接Ｗｅｂ情報をアクセスするＷｅｂページ生成して送信することができる。

【００３０】

更に、本発明は、請求項９に記載されるように、画像を形成する画像形成手段と、前記画像形成手段を制御する画像形成制御手段とを有し、前記Ｗｅｂ情報生成手段と前記他のＷｅｂ情報生成手段との少なくとも１つのＷｅｂ情報生成手段は、前記画像形成制御手段から前記画像形成手段に関する情報を取得して、その取得した情報の基づいて前記Ｗｅｂ情報を生成するように構成することができる。

【００３１】

このような情報処理装置において、端末種別情報を継承しつつ、情報処理装置に実装される画像形成するプロッタ又はスキャナ等の機器のステータスをＷｅｂ情報として提供することができる。

【００３２】

また、本発明は、請求項１０に記載されるように、前記表示情報生成手段は、前記基準パスに基づいて共通に設定される前記端末種別情報に基づいて、前記Ｗｅｂ情報生成手段又は前記他のＷｅｂ情報生成手段によって生成された前記Ｗｅｂ情報又は前記他のＷｅｂ情報にイメージを付加した前記Ｗｅｂページを生成するように構成することができる。

【００３３】

このような情報処理装置において、端末の表示画面が例えばＰＣより小さい場

合、イメージの表示を行わないようにＷｅｂページを生成することができる。また、端末の表示画面がＰＣより大きい場合、より大きいイメージ又は複数のイメージを表示するようにＷｅｂページを生成することができる。

【００３４】

更に、本発明は、請求項１１に記載されるように、前記表示情報生成手段は、前記基準パスに基づいて共通に設定される前記端末種別情報に基づいて、前記Ｗｅｂ情報生成手段又は前記他のＷｅｂ情報生成手段によって生成された前記Ｗｅｂ情報又は前記他のＷｅｂ情報を前記端末に対応した文字サイズで表示する前記Ｗｅｂページを生成するように構成することができる。

【００３５】

このような情報処理装置において、端末の表示画面が例えばＰＣより小さい場合、小さい文字サイズで表示するＷｅｂページを生成することができる。また、端末の表示画面がＰＣより大きい場合、大きい文字サイズで表示するＷｅｂページを生成することができる。

【００３６】

また、本発明は、請求項１２に記載されるように、前記表示情報生成手段は、前記基準パスに基づいて、前記Ｗｅｂ情報生成手段又は前記他のＷｅｂ情報生成手段によって生成された前記Ｗｅｂ情報又は前記他のＷｅｂ情報を前記端末に対応した文字数で表示する前記Ｗｅｂページを生成するように構成することができる。

【００３７】

このような情報処理装置において、端末の表示画面が例えばＰＣより小さい場合、少ない文字数で表示するＷｅｂページを生成することができる。また、端末の表示画面がＰＣより大きい場合、文字数を多くして詳細にしたＷｅｂページを生成することができる。

【００３８】

更に、前記課題を解決するための手段として、本発明は、上記情報処理装置における処理をコンピュータに行なわせるための情報処理方法とすることもできる。

【００３９】

【発明の実施の形態】

以下、本発明の実施の形態を図面に基づいて説明する。

【００４０】

本発明の一実施例に係る情報処理装置は、プリンタ、ＦＡＸ、コピー等の複数の異なる画像形成機能の少なくとも１つを有すると共に、複数のＷｅｂアプリケーションによって画像形成に関する情報を提供する。本実施例において、説明の便宜上、主に、印刷処理又は印刷を行うプロッタに関する情報が提供される場合について説明する。

【００４１】

図１は、本発明の一実施例に係る情報処理装置のハードウェア構成を示すブロック図である。図１において、情報処理装置１００は、コンピュータによって制御される装置であって、ＣＰＵ（中央処理装置）１１と、ＲＯＭ（Read-Only Memory）１２と、ＲＡＭ（Random Access Memory）１３と、不揮発性ＲＡＭ（non-volatile Random Access Memory）１４と、リアルタイムクロック１５、イーサネット（登録商標）Ｉ／Ｆ（Ethernet（登録商標） Interface）２１と、ＵＳＢ（Universal Serial Bus）２２と、ＩＥＥＥ（Institute of Electrical and Electronics Engineers）1284 ２３と、ハードディスクＩ／Ｆ２４と、エンジンＩ／Ｆ２５と、ＲＳ－２３２Ｃ　Ｉ／Ｆ２６とで構成され、システムバスＢに接続される。

【００４２】

ＣＰＵ１１は、ＲＯＭ１２に格納されたプログラムに従って情報処理装置１００を制御する。ＲＡＭ１３には、例えば、各インターフェース２１から２６に接続される資源に領域が割り当てられる。不揮発性ＲＡＭ１４には、情報処理装置１００を制御するためにＣＰＵ１１による処理で必要な情報が格納される。リアルタイムクロック１５は、現時刻を計ると共に、処理を同期させる場合にＣＰＵ１１によって使用される。

【００４３】

イーサネット（登録商標）Ｉ／Ｆ２１には、１０ＢＡＳＥ－Ｔ又は１００ＢＡ

ＳＥ－ＴＸ等のイーサネット（登録商標）用インターフェースケーブルが接続される。ＵＳＢ２２には、ＵＳＢ用インターフェースケーブルが接続される。ＩＥＥＥ1284　２３には、ＩＥＥＥ1284用インターフェースケーブルが接続される。

【００４４】

ハードディスクＩ／Ｆ２４には、ハードディスク３４が接続され、ネットワークを介して送信された印刷すべき文書の文書データ、又は、印刷処理後の画像データがハードディスクＩ／Ｆ２４を介してハードディスク３４に格納される。エンジンＩ／Ｆ２５には、文書データに基づいて所定媒体に印刷を行うプロッタ３５－１及び画像データを取り込むスキャナ３５－２等が接続される。ＲＳ－２３２Ｃ　Ｉ／Ｆ２６には、オペレーションパネル３６が接続され、ユーザへの情報の表示及びユーザから入力情報又は設定情報の取得が行われる。

【００４５】

次に、図１に示すようなハードウェア構成を有し、複数の異なる画像形成処理を可能とし、かつ、複数のＷｅｂアプリケーションを有する情報処理装置１００の機能構成について説明する。

【００４６】

図２は、情報処理装置の機能構成を示すブロック図である。図２において、情報処理装置１００は、インターネット１６を介して表示画面の利用形態の異なるクライアントＰＣ４１、携帯電話器４２及びＰＤＡ（Personal Digital Assistants）端末４３等の端末４０と接続可能であって、各端末４０からの要求に応じて、その要求に対する応答として情報を提供するコンピュータである。

【００４７】

情報処理装置１００は、主に、ネットワーク制御部１０１と、シーケンス制御ライブラリ１１０と、Ｗｅｂアプリ振分共通ライブラリ１２０と、Ｗｅｂページハンドラ２００と、ＳＯＡＰ(Simple Object Access Protocol)ライブラリ２０１と、ＸＭＬ(eXtensible Markup Language)ライブラリ２０３と、ＸＳＬＴ（XSL Transformations）プロセッサ２０５と、Ｗｅｂページ機能（ＷＰＦ）３００と、プリンタ制御部１０３と、スキャナ制御部１０５とを有する。

【００４８】

ネットワーク制御部１０１は、少なくともＨＴＴＰ（Hyper Text Transfer Protocol）による通信制御を行うＨＴＴＰデーモン２を有し、ＨＴＴＰに従って端末４０から要求を受信し、その要求に対する応答としてその要求に応じた情報提供を行う。

【００４９】

ＨＴＴＰデーモン２は、端末４０から要求を受信すると、複数のＷｅｂアプリケーション３０１において、ページ遷移した際にも端末４０の表示画面での利用形態に関する属性を示す属性情報が継承されるようにする所定の処理を実行するために、要求で指定されるＵＲＬ（Uniform Resource Locator）に仮の共通パス情報と、上記所定の処理を実行するためのＷｅｂアプリケーションＩＤとＣＧＩ(Common Gateway Interface)とを付加する。ページ遷移とは、表示されているＷｅｂページから、そのＷｅｂページにリンクされる他のＷｅｂページを表示させることを言う。

【００５０】

ＨＴＴＰデーモン２は、必要な情報を付加して、図３に示すようなＵＲＬを作成する。図３は、ＵＲＬ構成を示す図である。図３において、ＵＲＬは、大きく共通パス情報６０ａと相対パス情報６０ｂとで構成される。ＨＴＴＰデーモン２が端末４０から最初のページ要求を受信した際には、例えば、「http://AAA」のみがＵＲＬとして指定されており、このＵＲＬに対して、ＨＴＴＰデーモン２は、所定の「/TERMINAL/PROFILE/LANG/webDefaultApl/defaul.cgi」を付加する。

【００５１】

このようにして構成されたＵＲＬは、例えば、「http」等のアクセス手段を特定するプロトコル種別６１と、「AAA」で示されるＩＰアドレス６２と、「TERMINAL」で示される端末種別情報６３ａと、「PROFILE」で示されるプロファイル情報６３ｂと、「LANG」で示される言語情報６３ｃと、「webDefaultApl」で示されるＷｅｂアプリケーションＩＤ６４と、「default.cgi」で示されるページ情報６５とを有する。

【００５２】

そして、プロトコル種別６１と、ＩＰアドレス６２と、端末種別情報６３ａと

、プロファイル情報６３ｂと、言語情報６３ｃとが共通パス情報６０ａとして、以後のページ遷移において継承され、ページ遷移によって跨るＷｅｂアプリケーション３０１との間で参照される。また、ＷｅｂアプリケーションＩＤ６４と、ページ情報６５とが相対パス情報６０ｂを構成する。

【００５３】

図２に戻り、処理部の説明を続ける。

【００５４】

シーケンス制御ライブラリ１１０は、インターネット１６を介して行われるデータの送受信の処理シーケンスと各Ｗｅｂアプリケーション３０１とのデータの受け渡しの処理シーケンスとの違いを所定のシーケンス制御処理によって吸収する、複数のＷｅｂアプリケーション３０１に対して共通の処理部である。

【００５５】

Ｗｅｂアプリ振分共通ライブラリ１２０は、端末４０からの要求の解析及び端末４０への応答を生成し、複数のＷｅｂアプリケーション３０１に対して共通の処理部である。Ｗｅｂアプリ振分共通ライブラリ１２０は、Ｗｅｂページハンドラ２００にてＸＭＬで記述された応答を、ＸＳＬＴプロセッサ２０５によって各端末４０の端末種別に応じたＨＴＭＬ(HyperText Markup Language)による表示形式に変換する。

【００５６】

Ｗｅｂページハンドラ２００は、Ｗｅｂアプリケーション３０１が解釈可能な処理言語と、端末４０との間で行われる要求及び応答による通信制御で解釈される処理言語との変換を行う処理部である。Ｗｅｂページハンドラ２００は、要求に対応するＷｅｂアプリケーション３０１をＣＧＩを介して関数コールする。また、Ｗｅｂページハンドラ２００は、Ｗｅｂページ機能３００から通知された処理結果データをＸＭＬで記述するために処理結果データのシリアライズ要求をＳＯＡＰライブラリ２０１に対して行う。

【００５７】

ＳＯＡＰライブラリ２０１は、Ｃ言語の変数で与えられた処理結果データをＸＭＬによって記述することによってデータ変換をしてシリアライズする。本実施

例において、シリアライズするとは、ＸＭＬによってＷｅｂページ機能３００から通知された処理結果を記述することである。

【００５８】

ＸＭＬライブラリ２０３は、ＳＯＡＰライブラリ２０１に利用されることによってＸＭＬで処理結果をシリアライズする。また、ＸＭＬライブラリ２０３は、ＸＳＬＴプロセッサ２０５に利用されることによって、処理結果を示すＨＴＭＬを生成する。

【００５９】

ＸＳＬＴプロセッサ２０５は、端末４０の要求で指定されるＷｅｂアプリケーション３０１に対応するＸＳＬ(eXtensible Stylesheet Language)に基づいて、ＸＭＬライブラリ２０３を利用することによってＸＭＬで記述された処理結果をＨＴＭＬの記述に変換する。

【００６０】

Ｗｅｂページ機能３００は、Ｗｅｂページハンドラ２００から関数コールされると、例えば、情報処理装置１００に備えられた画像形成を制御するプリンタ制御部１０３又はスキャナ制御部１０５にＡＰＩ（Application Program Interface）１７を介して問い合わせて、プロッタ３５－１の状態又はスキャナ３５－２の状態等を示す情報を取得し、その情報をＷｅｂページハンドラ２００へ返す。また、Ｗｅｂページ機能３００は、同様にして、ＡＰＩ１７を介してネットワーク制御部１０１へ問い合わせることによって取得した情報をＷｅｂページハンドラ２００へ返す。

【００６１】

次に、Ｗｅｂアプリケーションをするための詳細な構成を図４及び図５で説明する。図４は、Ｗｅｂアプリケーションの実行を説明するためのブロック図である。図５は、ハッシュリストを示す図である。図４において、Ｗｅｂページハンドラ２００は、例えば、図３に示すようなＵＲＬの構成に基づいた情報提供が行えるように、端末４０からの最初の要求時に実行されるデフォルトハンドラ２１０と、端末４０からの指示によるページ遷移によって実行されるネットワーク設定ハンドラ２２０と、同様にページ遷移によって実行されるシステム構成情報提

供ハンドラ２３０等を有する。

【００６２】

Ｗｅｂページ機能３００は、Ｗｅｂページハンドラ２００の各ハンドラに対応して、Ｗｅｂデフォルトアプリ３１０と、Ｗｅｂネットワーク設定アプリ３２０と、Ｗｅｂシステム構成情報提供アプリ３３０等の複数のＷｅｂアプリケーション３０１を有する。

【００６３】

端末４０に初期画面が表示されるまでに、デフォルトハンドラ２１０は、図３に示されるＵＲＬの形式に従って指定されるページ情報６５に基づいて、順に、「default.cgi」、「mainFrame.cgi」、「menuPage.cgi」、「topPage.cgi」、「headerPage.cgi」を介してＷｅｂデフォルトアプリ３１０を関数コールする。その結果、例えば、情報処理装置１００に実装されたプロッタ３５－１又はスキャナ３５－２等の機器の状態が端末４０に表示される。

【００６４】

ネットワーク設定ハンドラ２２０は、端末４０のユーザによって入力されたネットワーク設定情報を情報処理装置１００内に設定するためのＷｅｂネットワーク設定アプリ３２０を関数コールする。また、システム構成情報提供ハンドラ２３０は、Ｗｅｂシステム構成情報提供アプリ３３０を関数コールすることによって、端末４０からの要求に応じて情報処理装置１００によって成されるシステムの構成情報を取得する。

【００６５】

Ｗｅｂアプリ振分共通ライブラリ１２０は、ＵＲＬからＷｅｂアプリケーションＩＤ６４を取得し、図５（Ａ）に示されるハッシュリスト５０ａを参照して、そのＷｅｂアプリケーションＩＤ６４が「webDefaultApl」を示す場合、「default_handler」を実行する。Ｗｅｂアプリケーションｉｄ６４が他のＷｅｂアプリケーションＩＤを示す場合についても同様に、ハッシュリスト５０ａを参照して、実行すべきＷｅｂアプリケーションに対応するハンドラを実行する。

【００６６】

また、Ｗｅｂページハンドラ２００の各ハンドラ２１０、２２０及び２３０は

、指定されたＵＲＬからページ情報６５を取得し、図５（Ｂ）に示されるハッシュリスト５０ｂを参照して、対応する関数をコールする。例えば、ページ情報６５が「topPage.cgi」を示す場合、ハッシュリスト５０ｂから「topPage()」をコールすべき関数として取得する。ページ情報６５が他のＣＧＩを示す場合についても同様に、ハッシュリスト５０ｂを参照して、コールすべき関数を取得する。

【００６７】

更に、Ｗｅｂアプリ振分共通ライブラリ１２０は、ＸＳＬＴプロセッサ２０５に対してＸＭＬで記述された処理結果をＨＴＭＬに変換するＸＳＬ変換要求を行うために、Ｗｅｂページ機能ＩＤとして指定されたＵＲＬからページ情報６５を取得し、図５（Ｂ）に示されるハッシュリスト５０ｂを参照して、対応するスタイルシートを取得する。例えば、ページ情報６５が「topPage.cgi」を示す場合、Ｗｅｂアプリ振分共通ライブラリ１２０は、ハッシュリスト５０ｂから「topPage.xsl」をＸＳＬ変換に必要なスタイルシートとして取得する。ページ情報６５が他のＣＧＩを示す場合についても同様に、ハッシュリスト５０ｂを参照して、ＸＳＬ変換に必要なスタイルシートを取得する。

【００６８】

図４及び図５（Ｂ）において、Ｗｅｂページ機能ＩＤとしてのＵＲＬのページ情報６５が「default.cgi」である場合、default()関数が実行され、端末４０の端末種別が決定され、ＨＴＴＰデーモン２によって仮に作成されたＵＲＬの端末種別情報にその決定した端末種別が設定された初期画面が作成される。

【００６９】

ページ情報６５が「mainFrame.cgi」である場合、mainFrame()関数が実行され、初期画面を構成するフレーム毎に、呼び出すべき所定のＣＧＩが設定される。例えば、mainFrame()関数によって、メニューページを表示するフレームと、トップページを表示するフレームと、ヘッダページを表示するフレームとが構成されるような情報が生成される。

【００７０】

ページ情報６５が「menuPage.cgi」である場合、menuPage()関数が実行され、メニューページを作成するための情報が生成される。ページ情報６５が「topPag

e.cgi」である場合、topPage()関数が実行され、例えば、プロッタ３５－１の状態がプリンタの状態として表示するトップページが作成するための情報が生成される。ページ情報６５が「headerPage.cgi」である場合、headerPage()関数が実行され、ヘッダページを作成するための情報が生成される。

【００７１】

また、ページ情報６５が「netsetting.cgi」である場合、setting()関数が実行され、システム設定を行うためのページを作成するための情報が生成される。ページ情報６５が「sysconfig.cgi」である場合、config()関数が実行され、システム構成を示す情報が生成される。

【００７２】

説明の便宜上、ＷｅｂアプリケーションＩＤの名称とＷｅｂページハンドラの名称とを統一させているが、このような名称に限定されるものではなく、図５（Ａ）のようなハッシュリスト５０ａによって対応付けがなされれば良い。同様に、Ｗｅｂページ機能ＩＤと、関数と、スタイルシートの名称を統一させているが、このような名称に限定されるものではなく、図５（Ｂ）のようなハッシュリスト５０ｂによって対応付けがなされれば良い。

【００７３】

次に、端末４０からの最初の要求に対して情報提供が行われるまでの処理フローについて説明する。図６及び図７は、ｄｅｆａｕｌｔ．ｃｇｉによる処理フローを示す図である。

【００７４】

図６において、端末４０は、ＨＴＴＰのＧＥＴメソッドを使用して、ユーザエージェント情報を付加した要求を行う（ステップＳ１１）。端末４０は、例えば、ＵＲＬ「http://AAA」を指定することによってその要求を行う。

【００７５】

この要求を受信した情報処理装置１００のＨＴＴＰデーモン２は、ＵＲＬ「http://AAA」に端末種別情報６３ａと、プロファイル情報６３ｂと、言語情報６３ｃと、ＷｅｂアプリケーションＩＤ６４と、ページ情報６５とを付加してシーケンス制御ライブラリ１１０へこの要求を通知する（ステップＳ１２）。例えば、

ＨＴＴＰデーモン２によって、端末種別情報６３ａとして「ＴＥＲＭＩＮＡＬ」、プロファイル情報６３ｂとして「ＰＲＯＦＩＬＥ」、言語情報６３ｃとして「ＬＡＮＧ」、ＷｅｂアプリケーションＩＤ６４として「ｗｅｂＤｅｆａｕｌｔＡｐｌ」、ページ情報６５として「ｄｅｆａｕｌｔ．ｃｇｉ」がＵＲＬ「http://AAA」に付加され、「http://AAA/TERMINAL/PROFILE/LANG/webDefaultApl/default.cgi」がＵＲＬとして構成される。

【００７６】

シーケンス制御ライブラリ１１０は、Ｗｅｂアプリ振分共通ライブラリ１２０を呼び出す（ステップＳ１３）。その際、シーケンス制御ライブラリ１１０は、端末４０との接続に関する情報を示すＨＴＴＰ接続情報と、ハッシュリスト５０ａ及び５０ｂ（以下、ハッシュリスト５０ａ及び５０ｂを総称してハッシュリストと言う）とを引数として設定する。このＨＴＴＰ接続情報は、接続が切断されるまで端末４０との接続を一意に特定する。

【００７７】

Ｗｅｂアプリ振分共通ライブラリ１２０は、ＨＴＴＰデーモン２によって生成されたＵＲＬに基づいて、「ｗｅｂＤｅｆａｕｌｔＡｐｌ」を示すＷｅｂアプリケーションＩＤ６４に対応するＷｅｂページハンドラ２００として、ハッシュリスト５０ａからデフォルトハンドラ２１０（「default_handler」）を特定して呼び出す（ステップＳ１４）。

【００７８】

Ｗｅｂページハンドラ２００では、デフォルトハンドラ２１０が、ＵＲＬから「default.cgi」を示すページ情報６５を取得し、その「default.cgi」に対応するＷｅｂページ機能３００として、ハッシュリスト５０ｂからdefault()関数を特定して呼び出す（ステップＳ１５）。この際、ＨＴＴＰ接続情報が引数として設定される。例えば、Ｗｅｂページ機能３００がＣ言語で開発されている場合、Ｗｅｂページハンドラ２００は、Ｃ言語の処理型に変換してdefault()関数をコールする。

【００７９】

default()関数の実行によって、ＨＴＴＰ接続情報に基づいて、端末情報が解

析され、決定した共通パス情報６０ａによって初期画面を呼び出すＵＲＬが作成される（ステップＳ１６）。そのＵＲＬを含む処理結果データとしてＷｅｂページハンドラ２００におけるデフォルトハンドラ２１０へ通知される（ステップＳ１７）。default.cgiによる処理では、共通パス情報６０ａを決定し、決定した共通パス情報６０ａと初期画面のフレーム構成を決定するための相対パス情報６０ｂとによってＵＲＬが設定される。例えば、共通パス情報６０ａとして「/pc/admin/ja」が決定され、相対パス情報６０ｂとして、初期画面をフレームによって幾つかの表示領域に分割する「/webDefaultApl/mainFrame.cgi」を設定する。

【００８０】

デフォルトハンドラ２１０は、処理結果データをＸＭＬで記述するために、シリアライズ要求をＳＯＡＰライブラリ２０１に対して行う（ステップＳ１８）。ＳＯＡＰライブラリ２０１は、例えば、Ｃ言語の構造体で示される処理結果データに基づいて、ＤＯＭ（Document Object Model）を作成し、必要な要素及び属性を追加して（ステップＳ１９）、ＸＭＬライブラリ２０３によって処理結果データをＸＭＬで記述する（ステップＳ２０）。ＸＭＬで記述された処理結果データは、ＳＯＡＰライブラリ２０３によってシリアライズ結果としてデフォルトハンドラ２１０に通知される（ステップＳ２１）。Ｗｅｂページハンドラ２００において、デフォルトハンドラ２１０は、通知されたシリアライズ結果を端末４０に表示すべき表示データ「Response DOM」として、Ｗｅｂアプリ振分共通ライブラリ１２０へ通知する（ステップＳ２２）。

【００８１】

Ｗｅｂアプリ振分共通ライブラリ１２０は、ＸＭＬで記述される表示データを端末４０の種別に応じたＨＴＭＬの記述に変換するために、ハッシュリスト５０ｂを参照して、「default.cgi」を示すページ情報５６に対応するスタイルシート「default.xsl」を取得して、ＸＳＬＴプロセッサ２０５に対してＸＳＬ変換要求を行う（ステップＳ２３）。この際、表示データ「Response DOM」と、スタイルシート「default.xsl」とが引数として設定される。

【００８２】

ＸＳＬ変換要求を受信したＸＳＬＴプロセッサ２０５は、引数で指定されたス

タイルシート「default.xsl」を実行することによって、ＸＭＬで記述された表示データ「Response DOM」の構文を解析し（ＸＭＬパース）、ＤＯＭを作成し、要素及び属性を追加して（ステップＳ２４）、ＸＭＬライブラリ２０３によってＨＴＭＬの記述に変換する（ステップＳ２５）。ＸＳＬＴプロセッサ２０５は、変換したＨＴＭＬの記述をＸＳＬ変換結果としてＷｅｂアプリ振分共通ライブラリ１２０へ通知する（ステップＳ２６）。

【００８３】

このように、ページ情報６５が「default.cgi」の場合のＷｅｂページ機能３００によってＨＴＭＬで記述された処理結果が、Ｗｅｂアプリ振分共通ライブラリ１２０にデフォルトＨＴＭＬとして出力される。デフォルトＨＴＭＬでは、ＨＴＭＬで「mainFrame.cgi」が呼び出されるように記述され、デフォルトＨＴＭＬ応答として出力される。

【００８４】

このデフォルトＨＴＭＬ応答は、順に、Ｗｅｂアプリ振分共通ライブラリ１２０からシーケンス制御ライブラリ１１０へ通知され（ステップＳ２７）、シーケンス制御ライブラリ１１０からＨＴＴＰデーモン２へ通知される（ステップＳ２８）。そして、ＨＴＴＰデーモン２によって、ＨＴＴＰに従ってデフォルトＨＴＭＬ応答がステップＳ１１における端末４０の要求に対する応答として送信され（ステップＳ２９）、default.cgiによってＷｅｂデフォルトアプリ３１０を実行する処理フローを終了する。

【００８５】

ステップＳ１７では、共通パス情報６０ａを決定し、決定した共通パス情報６０ａを相対パス情報６０ｂに付加するのみであるので、Ｗｅｂデフォルトアプリ３１０にて処理結果をＨＴＭＬで記述し、そのＨＴＭＬで記述された処理結果を出力するようにし、ステップＳ１８からＳ２６を省略しても良い。

【００８６】

次に、図６のステップＳ１６にてdefault()関数のコールによって実行される端末情報の解析処理について図８から図１１で詳述する。図８は、端末情報の解析処理について説明するためのフローチャート図である。図９は、ＨＴＴＰ接続

情報のデータ構造を示す図である。図１０は、ＨＴＴＰ要求情報のデータ構造を示す図である。図１１は、ヘッダ情報リストを示す図である。図９から図１１に示されるデータ構造は、図８に示される端末情報の解析処理にて参照されるデータ構造である。

【００８７】

図８での解析処理を説明する前に、図９から図１１に示される各データ構造について説明する。図９において、ＨＴＴＰ接続情報３４０は、接続毎に生成される情報であって、「HTTP_REQUEST_INFO　*request」にて示される記述３４６は端末４０からの要求の内容へのポインタを示し、「HTTP_RESPONSE_INFO　*response」にて示される記述３４７は端末４０への応答の内容へのポインタを示す。この記述３４６によって指定されるポインタの先には、ＨＴＴＰ要求情報が図１０に示すように格納されている。

【００８８】

図１０において、ＨＴＴＰ要求情報３５０は、リクエストをした側の端末４０のアドレスを示すリモートアドレス３５１と、そのリクエストを受信した側のホストアドレスを示すローカルアドレス３５２と、ＰＯＳＴ、ＧＥＴ等のリクエストメソッド３５３と、リクエストのＵＲＬを示すリクエストＵＲＬ３５４と、リクエストのＨＴＴＰバージョン３５５と、ヘッダ情報のリスト３６０と、エンコードされた情報のリスト３５７とを有する。

【００８９】

リクエストのＵＲＬ３５４には、図３に示されるような共通パス情報６０ａと相対パス情報６０ｂとで構成されるＵＲＬがシーケンス制御ライブラリ１１０によって設定される。ＨＴＴＰ要求情報３５０は、ＨＴＴＰ接続情報３４０がＷｅｂページハンドラ２００とＷｅｂページ機能３００とへ通知されることによって、同時に通知される。従って、各Ｗｅｂページハンドラ２００と各Ｗｅｂアプリケーション３０１は、共通パス情報６０ａを参照できる。

【００９０】

そして、ヘッダ情報リスト３６０は、例えば、図１１に示すような情報によって構成される。図１１において、ヘッダ情報リスト３６０は、「User-Agent:Moz

illa/4.0{compatible;MSIE 6.0;Windows（登録商標） NT 5.0;Q312461;NET CLR 1.0.3705}にて示されるユーザエージェント情報３６５を有する。このユーザエージェント情報３６５は、端末４０の種別を示し、この記述を解析することによって、端末４０での表示画面に応じた情報提供をすることができる。

【００９１】

図８において、default()関数のコールによって、Ｗｅｂデフォルトアプリ３１０は、ＨＴＴＰ接続情報３４０からＨＴＴＰ要求情報３５０を取得する（ステップＳ１）。更に、Ｗｅｂデフォルトアプリ３１０は、ＨＴＴＰ要求情報３５０からユーザエージェント情報３６５を取得する（ステップＳ２）。そして、Ｗｅｂデフォルトアプリ３１０は、ユーザエージェント情報３６５のＷｅｂブラウザ種別に基づいて、Ｗｅｂブラウザ種別―端末種別対応表（図１２）を参照して、端末４０の端末種別を特定する（ステップＳ３）。

【００９２】

Ｗｅｂデフォルトアプリ３１０は、Ｗｅｂブラウザ種別で端末種別が判断できたか否かを判断する（ステップＳ４）。判断できる場合、ステップＳ３８へ進む。判断できない場合、ユーザエージェント情報３６５のＷｅｂブラウザＯＳに基づいて、Ｗｅｂブラウザ０Ｓ－端末種別対応表を参照して、端末４０の端末種別を特定する（ステップＳ５）。

【００９３】

Ｗｅｂデフォルトアプリ３１０は、Ｗｅｂブラウザ０Ｓで端末種別が判断できたか否かを判断する（ステップＳ６）。判断できない場合、サポート対象外であることを示すエラー情報を出力し（ステップＳ７）、端末情報の解析処理を終了する。ステップＳ７において、Ｗｅｂデフォルトアプリ３１０は、サポート対象外であることを示すエラー情報を出力する代わりに、所定の端末種別を端末４０の端末種別として設定し、ステップＳ８へ進むようにしても良い。

【００９４】

一方、ステップＳ４又はＳ６にて、端末種別が判断できる場合、Ｗｅｂデフォルトアプリ３１０は、ＵＲＬの端末種別情報６３ａに端末種別を設定し（ステップＳ８）、端末情報の解析処理を終了する。

【００９５】

図１２は、Ｗｅｂブラウザ種別―端末種別対応表を示す図である。図１２において、Ｗｅｂブラウザ種別―端末種別対応表３７０は、Ｗｅｂブラウザ種別と端末種別等の項目を有し、Ｗｅｂブラウザ種別に応じた端末種別が管理される。

【００９６】

Ｗｅｂブラウザ種別として、例えば、「ＤＯＣＯＭＯ（登録商標）」、「ｘｉｎｏ（登録商標）」、「ＭＳＰＩＥ（登録商標）」、「ＨａｎｄＨＴＴＰ（登録商標）」、「ＭＳＩＥ（登録商標）」、「Ｍｏｚｉｌｌａ（登録商標）」、「ｍｏｚｉｌｌａ／５（登録商標）」、「Ｎｅｔｓｃａｐｅ６（登録商標）」等が管理される。

【００９７】

端末種別として、「ＩＭＯＤＥ（登録商標）」、「ＰＤＡ」、「０」、「ＰＣ」等がある。Ｗｅｂブラウザ種別「ＤＯＣＯＭＯ（登録商標）」に対応する端末種別は「ＩＭＯＤＥ（登録商標）」であり、Ｗｅｂブラウザ種別「ｘｉｉｎｏ（登録商標）」、「ＭＳＰＩＥ（登録商標）」、「ＨａｎｄＨＴＴＰ（登録商標）」に対応する端末種別は「ＰＤＡ」であり、Ｗｅｂブラウザ種別「Ｍｏｚｉｌｌａ（登録商標）」、「Ｍｏｚｉｌｌａ（登録商標）」、「ｍｏｚｉｌｌａ／５（登録商標）」、「Ｎｅｔｓｃａｐｅ６（登録商標）」に対応する端末種別は「ＰＣ」である。端末種別は、上記以外に限定されるものではなく、更に多くの種別があっても良い。

【００９８】

端末種別「０」は、端末種別を特定できないことを示す。図８のステップＳ４での判断処理において、判断できない場合に相当する。例えば、ＩｎｔｅｒｎｅｔＥｘｐｌｏｅｒ（登録商標）からの要求である場合、通常、端末種別が「ＰＤＡ」の場合にはユーザエージェント情報３６５にＷｅｂブラウザ種別「ＭＳＰＩＥ」が設定され、端末種別が「ＰＣ」の場合にはユーザエージェント情報３６５にＷｅｂブラウザ種別「ＭＳＩＥ」が設定される。しかしながら、端末種別が「ＰＤＡ」の場合であっても、ユーザエージェント情報３６５にＷｅｂブラウザ種別「ＭＳＩＥ」が設定されることがある。従って、Ｗｅｂブラウザ種別「ＭＳＩ

Ｅ」に対応する端末種別を予め特定することができないため、「０」が設定されている。

【００９９】

このように端末種別を特定できない場合、図１３に示すような対応表が参照される。図１３は、ＷｅｂブラウザＯＳ－端末種別対応表を示す図である。図１３において、ＷｅｂブラウザＯＳ－端末種別対応表３８０は、ＷｅｂブラウザＯＳと端末種別等の項目を有し、ＷｅｂブラウザＯＳに応じた端末種別が管理される。

【０１００】

ＷｅｂブラウザＯＳとして、例えば、「Windows（登録商標）３．１」、「Windows（登録商標）９５」、「Windows（登録商標）９８」、「Windows（登録商標）ＭＥ」、「Windows（登録商標）ＮＴ」、「Windows（登録商標）２０００」、「Windows（登録商標）ＸＰ」、「Ｍａｃ　６８Ｋ（登録商標）」、「ＭａｃＰｏｗｅｒＰＣ（登録商標）」、「Ｓｏｌａｒｉｓ（登録商標）」、「Ｌｉｎｕｘ（登録商標）」、「ＦｒｅｅＢＳＤ（登録商標）」、「ＡＩＸ（登録商標）」、「ＩＲＩＸ（登録商標）」、「ＨＰ（登録商標）」、「ＯＳ／２（登録商標）」、「Windows（登録商標）ＣＥ」等が管理される。

【０１０１】

端末種別として、「ＰＤＡ」、「ＰＣ」等がある。携帯電話器４２は、端末種別「ＰＤＡ」に含む。図１３に示すＷｅｂブラウザＯＳ－端末種別対応表３８０において、ＷｅｂブラウザＯＳ「Windows（登録商標）３．１」から「ＯＳ／２（登録商標）」に対応する端末種別は「ＰＣ」であり、ＷｅｂブラウザＯＳ「Windows（登録商標）ＣＥ」に対応する端末種別は「ＰＤＡ」である。

【０１０２】

このようにＷｅｂブラウザ種別―端末種別対応表３７０とＷｅｂブラウザＯＳ―端末種別対応表３８０とによって端末種別を特定することができる。

【０１０３】

ｄｅｆａｕｌｔ．ｃｇｉによるデフォルトＨＴＭＬ出力について図１４で説明する。図１４は、デフォルトＨＴＭＬ出力の例を示す図である。図１４において

、デフォルトＨＴＭＬ４００は、端末４０にて表示されることはなく、Ｊａｖａ（登録商標）Ｓｃｒｉｐｔ（登録商標）の記述４０１によって、トップページを呼び出すためのスクリプトが実行される。

【０１０４】

「self.document.cookie=”cookieOnOffchecker=on; path=/”」で示される記述４０２によって、端末４０のＷｅｂブラウザの情報をｃｏｏｋｉｅによって取得し、「self.location.pathname=”/pc/admin/ja/webDefaultApl/mainframe.cgi”」で示される記述４０３によって、端末種別「ＰＣ」を継承して、Ｗｅｂページ機能３００のＷｅｂデフォルトアプリ３１０が実行される、つまり、「mainframe.cgi」によってメインフレームＨＴＭＬを要求する。記述４０３によって、以後ユーザによって行われる一連のページ遷移において、<a href>タグで指定される相対パスの基準となる絶対パスが指定されたことになる。

【０１０５】

尚、インターネット１６を介してクライアントＰＣ４１に情報が提供される場合、画面をフレームによって複数に分割することがよく行われる。図１４では、そのような場合のデフォルトＨＴＭＬ出力の例を示している。一方、フレームによって画面を分割しないで初期画面を表示する場合は、例えば、記述４０３は、「self.location.pathname=”/pc/admin/ja/webDefaultApl/topPage.cgi”」のように記述され、提供されるべき情報を表示するためのトップページが直接呼ばれるように設定される。

【０１０６】

以下、画面がフレームによって分割されるような場合に端末４０の端末種別に応じた情報提供が行われる処理について説明をする。

【０１０７】

以後、ページ遷移が行われた際には、相対パス情報６０ｂのみが指定されるため、共通パス情報６０ａに含まれる端末種別「ＰＣ」が常に継承されることになる。つまり、同一の共通パス情報６０ａを複数のＷｅｂアプリケーション３０１が参照する。

【０１０８】

図１５及び図１６は、ｍａｉｎＦｒａｍｅ．ｃｇｉによる処理フローを示す図である。図１５において、端末４０は、メインフレームＨＴＭＬを要求する（メインフレームＨＴＭＬ要求）（ステップＳ３１）。「/pc/admin/ja/webDefaultApl/mainFrame.cgi」が指定され、ｆｏｒｍデータが添付されたＧＥＴメソッドでメインフレームＨＴＭＬ要求が行われる。

【０１０９】

情報処理装置１００のＨＴＴＰデーモン２は、ＨＴＴＰ接続情報３４０とハッシュリストとをシーケンス制御ライブラリ１１０に通知する（ステップＳ３２）。

【０１１０】

この場合、ＨＴＴＰデーモン２によって、ＨＴＴＰ接続情報３４０と関連付けされるＨＴＴＰ要求情報３５０のリクエストメソッドには「ＧＥＴ」が設定され、リクエストのＵＲＬ３５４には「/pc/admin/ja/webDefaultApl/mainFrame.cgi」が指定される。この指定によって、以後、端末４０から相対パスによって要求を受信した場合、「/pc/admin/ja」を示す共通パス情報６０ａが参照可能となる。従って、端末４０の端末種別が「ＰＣ」であることが分かる。また、ＨＴＴＰデーモン２によって、ハッシュリストに受信したｆｏｒｍデータが設定される。

【０１１１】

シーケンス制御ライブラリ１１０は、Ｗｅｂアプリ振分共通ライブラリ１２０を呼び出す（ステップＳ３３）。その際、シーケンス制御ライブラリ１１０は、端末４０との接続に関する情報を示すＨＴＴＰ接続情報と、ハッシュリストとを引数として設定する。

【０１１２】

Ｗｅｂアプリ振分共通ライブラリ１２０は、ＵＲＬに基づいて、「ｗｅｂＤｅｆａｕｌｔＡｐｌ」を示すＷｅｂアプリケーションＩＤ６４に対応するＷｅｂページハンドラ２００として、ハッシュリスト５０ａからデフォルトハンドラ２１０（「default_handler」）を特定して呼び出す（ステップＳ３４）。

【０１１３】

Ｗｅｂページハンドラ２００では、デフォルトハンドラ２１０が、ＵＲＬから

「mainFrame.cgi」を示すページ情報６５を取得し、その「mainFrame.cgi」に対応するＷｅｂページ機能３００として、ハッシュリスト５０ｂからmainFrame()関数を特定して呼び出す（ステップＳ３５）。この際、ＨＴＴＰ接続情報が引数として設定される。例えば、Ｗｅｂページ機能３００がＣ言語で開発されている場合、Ｗｅｂページハンドラ２００は、Ｃ言語の処理型に変換してmainFrame()関数をコールする。

【０１１４】

mainFrame()関数の実行によって、初期画面（topPage.cgi）を構成するフレーム情報が作成される（ステップＳ３６）。そのフレーム情報が処理結果データとして、Ｗｅｂページハンドラ２００におけるデフォルトハンドラ２１０へ通知される（ステップＳ３７）。

【０１１５】

以下、ステップＳ３８、Ｓ３９、Ｓ４０、Ｓ４１及びＳ４２によるＸＭＬで記述するシリアライズの処理は、上記ステップＳ１８、Ｓ１９、Ｓ２０、Ｓ２１及びＳ２２での処理と同様であるのでその説明を省略する。

【０１１６】

Ｗｅｂアプリ振分共通ライブラリ１２０は、ＸＭＬで記述される表示データを端末４０の種別に応じたＨＴＭＬの記述に変換するために、ハッシュリスト５０ｂを参照して、「mainFrame.cgi」を示すページ情報５６に対応するスタイルシート「mainFrame.xsl」を取得して、ＸＳＬＴプロセッサ２０５に対してＸＳＬ変換要求を行う（ステップＳ４３）。この際、表示データ「Response DOM」と、スタイルシート「mainFrame.xsl」とが引数として設定される。

【０１１７】

ＸＳＬ変換要求を受信したＸＳＬＴプロセッサ２０５は、引数で指定されたスタイルシート「mainFrame.xsl」を実行することによって、端末種別に応じたテンプレートをスタイルシートの記述から選択し、ＸＭＬで記述された表示データ「Response DOM」の構文を解析し（ＸＭＬパース）、ＤＯＭを作成し、要素及び属性を追加して（ステップＳ４４）、ＸＭＬライブラリによってＨＴＭＬの記述に変換する（ステップＳ４５）。端末種別情報６３ａが「ＰＣ」を示す場合、画

面をフレームによって複数の画面に分割するＨＴＭＬが出力される。一方、端末種別情報６３ａが「ＰＤＡ」を示す場合、直接トップページを呼び出すためのＪａｖａ（登録商標）Ｓｃｒｉｐｔ（登録商標）が記述されたＨＴＭＬが出力される。

【０１１８】

ＸＳＬＴプロセッサ２０５は、変換したＨＴＭＬの記述をＸＳＬ変換結果としてＷｅｂアプリ振分共通ライブラリ１２０へ通知する（ステップＳ４６）。

【０１１９】

ページ情報６５が「mainFrame.cgi」の場合のＷｅｂページ機能３００によってＨＴＭＬで記述された処理結果が、Ｗｅｂアプリ振分共通ライブラリ１２０にメインフレームＨＴＭＬとして出力される。メインフレームＨＴＭＬでは、メニューページを構成するフレームからは「menuPage.cgi」が呼び出され、トップページを構成するフレームからは「topPage.cgi」が呼び出され、ヘッダページを構成するフレームからは「headerPage.cgi」が呼び出されるように、ＨＴＭＬで記述される。

【０１２０】

このメインフレームＨＴＭＬ応答は、順に、Ｗｅｂアプリ振分共通ライブラリ１２０からシーケンス制御ライブラリ１１０へ通知され（ステップＳ４７）、シーケンス制御ライブラリ１１０からＨＴＴＰデーモン２へ通知される（ステップＳ４８）。そして、ＨＴＴＰデーモン２によって、ＨＴＴＰに従ってメインフレームＨＴＭＬ応答がステップＳ３１における端末４０の要求に対する応答として送信され（ステップＳ４９）、mainFrame.cgiによって実行される処理フローを終了する。

【０１２１】

端末４０がクライアントＰＣ４１である場合のｍａｉｎＦｒａｍｅ．ｃｇｉによるメインフレームＨＴＭＬ出力について図１７で説明する。図１７は、ＰＣ用のメインフレームＨＴＭＬ出力の例を示す図である。図１７において、メインフレームＨＴＭＬ４１０は、記述４１１によって端末４０のＷｅｂブラウザ上に３つのフレームを作成し、各<frame>タグからページをＣＧＩで呼び出す。

【０１２２】

<frame noresize src="/webDefaultApl/headerPage.cgi" name="header" marginheight="0" marginwidth="0" scrolling="no">によって示される記述４１２は、フレーム内にヘッダページを表示する。「/webDefaultApl/headerPage.cgi」で示される記述４１３によって，

「webDefaultApl」に基づいてデフォルトハンドラ２１０が実行され、「headerPage.cgi」に基づいてＷｅｂデフォルトアプリ３１０のheaderPage()関数が実行され、ヘッダページがフレーム内に表示される。

【０１２３】

また、<frame src="/webDefaultApl/menuPage.cgi" name="menu"marginheight="0" marginwidth="0" scrolling="auto">によって示される記述４１４は、フレーム内にメニューページを表示する。「/webDefaultApl/menuPage.cgi」で示される記述４１５によって、「webDefaultApl」に基づいてデフォルトハンドラ２１０が実行され、「menuPage.cgi」に基づいてＷｅｂデフォルトアプリ３１０のmenuPage()関数が実行され、メニューページがフレーム内に表示される。

【０１２４】

更に、<frame src="/webDefaultApl/topPage.cgi" name="top" marginheight="0" marginwidth="0" scrolling="auto">によって示される記述４１６は、フレーム内にトップページを表示する。「/webDefaultApl/topPage.cgi」で示される記述４１７によって、「webDefaultApl」に基づいてデフォルトハンドラ２１０が実行され、「topPage.cgi」に基づいてＷｅｂデフォルトアプリ３１０のtopPage()関数が実行され、トップページがフレーム内に表示される。

【０１２５】

記述４１２、４１４及び４１６は、相対パス情報６０ｂを指定するのみであるから、共通パス情報６０ａがＵＲＬで指定されることになる。

【０１２６】

図１７において、端末４０が携帯電話器４２又はＰＤＡ端末４３である場合、フレームを構成せずに、ｔｏｐＰａｇｅ．ｃｇｉのみを呼び出すように構成される。例えば、ＨＴＭＬの記述は、図１４に示すデフォルトＨＴＭＬ出力と同様の

記述であるが、jumpToMainFrame()の代わりに、self.location.pathname=”/pda/admin/ja/webDefaultApl/topPage.cgi”によって直接ｔｏｐＰａｇｅ．ｃｇｉを呼び出すようなjumpToTopPage()が使用される。

【０１２７】

図１８及び図１９は、ｔｏｐＰａｇｅ．ｃｇｉによる処理フローを示す図である。図１８において、端末４０は、情報処理装置１００に備えられたプロッタ３５－１又はスキャナ３５－２のステータスを表示するＨＴＭＬを要求する（トップページＨＴＭＬ要求）（ステップＳ５１）。「topPage.cgi」が指定され、ｆｏｒｍデータが添付されたＧＥＴメソッドでトップページＨＴＭＬ要求が行われる。

【０１２８】

情報処理装置１００のＨＴＴＰデーモン２は、ＨＴＴＰ接続情報３４０とハッシュリストとをシーケンス制御ライブラリ１１０に通知する（ステップＳ５２）。この場合、ＨＴＴＰデーモン２によって、ＨＴＴＰ接続情報３４０と関連付けされるＨＴＴＰ要求情報３５０のリクエストメソッドには「ＧＥＴ」が設定され、リクエストのＵＲＬには「topPage.cgi」が指定される。また、ＨＴＴＰデーモン２によって、ハッシュリストに受信したｆｏｒｍデータを設定する。

【０１２９】

シーケンス制御ライブラリ１１０は、Ｗｅｂアプリ振分共通ライブラリ１２０を呼び出す（ステップＳ５３）。その際、シーケンス制御ライブラリ１１０は、端末４０との接続に関する情報を示すＨＴＴＰ接続情報３４０と、ハッシュリストとを引数として設定する。

【０１３０】

Ｗｅｂアプリ振分共通ライブラリ１２０は、ＵＲＬに基づいて、「ｗｅｂＤｅｆａｕｌｔａｇｅＡｐｌ」を示すＷｅｂアプリケーションＩＤ６４に対応するＷｅｂページハンドラ２００として、ハッシュリスト５０ａからデフォルトハンドラ２１０（「default_handler」）を特定して呼び出す（ステップＳ５４）。

【０１３１】

Ｗｅｂページハンドラ２００では、デフォルトハンドラ２１０が、ＵＲＬから

「topPage.cgi」を示すページ情報６５を取得し、その「topPage.cgi」に対応するＷｅｂページ機能３００として、ハッシュリスト５０ｂからtopPage()関数を特定して呼び出す（ステップＳ５５）。この際、ＨＴＴＰ接続情報３４０が引数として設定される。例えば、Ｗｅｂページ機能３００がＣ言語で開発されている場合、Ｗｅｂページハンドラ２００は、Ｃ言語の処理型に変換してtopPage()関数をコールする。

【０１３２】

topPage()関数の実行によって、Ｗｅｂデフォルトアプリ３１０は、初期画面（topPage.cgi）に機器（例えば、プリンタとしてのプロッタ３５－１）のステータスを表示するために、ＡＰＩ１７を介してネットワーク制御部１０１に対して、デバイス名、コメント等の情報を要求する（ステップＳ５６－２）。ネットワーク制御部１０１は、その要求に応じてデバイス名、コメント等の情報を通知する（ステップＳ５６－４）。

【０１３３】

Ｗｅｂページ機能３００のＷｅｂデフォルトアプリ３１０は、例えば、プリンタのステータスを、ＡＰＩ１７を介してプリンタ制御部１０３に対して要求する（プリンタステータス要求）（ステップＳ５６－６）。プリンタ制御部１０３は、プロッタ３５－１の状態をプリンタステータスとしてＷｅｂデフォルトアプリ３１０へ通知する（ステップＳ５６－８）。

【０１３４】

そして、Ｗｅｂデフォルトアプリ３１０は、そのプリンタステータスと、ＨＴＴＰ要求情報３５０のリクエストのＵＲＬ３４５から端末種別情報６３ａと、プロファイル情報６３ｂと、言語情報６３ｃとを、処理結果データとして、Ｗｅｂページハンドラ２００におけるデフォルトハンドラ２１０へ通知する（ステップＳ５７）。

【０１３５】

以下、ステップＳ５８、Ｓ５９、Ｓ６０、Ｓ６１及びＳ６２によるＸＭＬで記述するシリアライズの処理は、上記ステップＳ１８、Ｓ１９、Ｓ２０、Ｓ２１及びＳ２２での処理と同様であるのでその説明を省略する。

【０１３６】

ＷｅｂアプリＪ振分共通ライブラリ１２０は、ＸＭＬで記述される表示データを端末４０の種別に応じたＨＴＭＬの記述に変換するために、ハッシュリスト５０ｂを参照して、「topPage.cgi」を示すページ情報５６に対応するスタイルシート「topPage.xsl」を取得して、ＸＳＬＴプロセッサ２０５に対してＸＳＬ変換要求を行う（ステップＳ６３）。この際、表示データ「Response DOM」と、スタイルシート「topPage.xsl」とが引数として設定される。

【０１３７】

ＸＳＬ変換要求を受信したＸＳＬＴプロセッサ２０５は、引数で指定されたスタイルシート「topPage.xsl」を実行することによって、端末種別に応じたテンプレートをスタイルシートの記述から選択し、ＸＭＬで記述された表示データ「Response DOM」の構文を解析し（ＸＭＬパース）、ＤＯＭを作成し、要素及び属性を追加して（ステップＳ６４）、ＸＭＬライブラリによってＨＴＭＬの記述に変換する（ステップＳ６５）。

【０１３８】

ＸＳＬＴプロセッサ２０５は、変換したＨＴＭＬの記述をＸＳＬ変換結果としてＷｅｂアプリ振分共通ライブラリ１２０へ通知する（ステップＳ６６）。

【０１３９】

ページ情報６５が「topPage.cgi」の場合のＷｅｂページ機能３００によってＨＴＭＬで記述された処理結果が、Ｗｅｂアプリ振分共通ライブラリ１２０にトップページＨＴＭＬとして出力される。トップページＨＴＭＬでは、プロッタ３５－１の状態に関する情報がＨＴＭＬで記述される。

【０１４０】

このトップページＨＴＭＬ応答は、Ｗｅｂアプリ振分共通ライブラリ１２０からシーケンス制御ライブラリ１１０へ通知され（ステップＳ６７）、シーケンス制御ライブラリ１１０からＨＴＴＰデーモン２へ通知される（ステップＳ６８）。そして、ＨＴＴＰデーモン２によって、ＨＴＴＰに従ってトップページＨＴＭＬ応答がステップＳ５１における端末４０の要求に対する応答として送信され（ステップＳ６９）、topPage.cgiによってを実行される処理フローを終了する。

【０１４１】

端末４０がクライアントＰＣ４１である場合、上記同様にして、各フレームから呼び出されたHeaderPage.cgi及びmenuPage.cgiによる処理が実行され、フレームによって分割された画面がクライアントＰＣ４１のＷｅｂブラウザに表示される。一方、端末４０が携帯電話器４２又はＰＤＡ端末４３である場合、フレームによる画面の分割は行われないので、一つの画面上にトップページのみが表示される。

【０１４２】

ＸＭＬで記述された処理結果について説明する。図２０は、ＸＭＬで記述されたｔｏｐＰａｇｅ．ｃｇｉによる処理結果の例を示す図である。

【０１４３】

図２０に示されるＸＭＬ４２０において、<networkResponse>から</networkResponse>までで示される記述４２１によってＷｅｂデフォルトアプリ３１０のtopPage()関数による処理結果が示される。<terminal>pc</terminal>で示される記述４２２は、端末４０がクライアントＰＣ４１であることを示し、<language>ja</language>で示される記述４２３は、端末４０のＷｅｂブラウザの言語が日本語であることを示し、<profile>admin</profile>で示される記述４２４は、端末４０の利用者が管理者であることを示している。記述４２２から４２４は、ＵＲＬの共通パス情報６０ａの端末種別情報６３ａ、プロファイル情報６３ｂ、言語情報６３ｃに基づく情報である。

【０１４４】

<deviceName>Printer 1</deviceName>で示される記述４２５は、機器名が「Printer 1」であることを示す。<comment>XXXXXXXXX</comment>で示される記述４２６は、「Printer 1」に関するコメントが「XXXXXXXXX」であることを示す。<status>Ready</status>で示される記述４２７は、「Printer 1」の状態が「正常」であることを示す。

【０１４５】

Ｗｅｂページハンドラ２００が、Ｗｅｂページ機能３００よる処理結果データをＸＭＬで記述してシリアライズする機能を有するため、Ｗｅｂページ機能３０

０の各Ｗｅｂアプリケーション３０１は、それぞれに端末種別に対応する処理部を有する必要がない。

【０１４６】

このようにＸＭＬで記述されシリアライズされた処理結果データを端末種別に基づいて、ＸＳＬＴプロセッサ２０５によってＨＴＭＬに変換するためのスタイルシートについて図２１で説明する。図２１は、処理結果をＸＭＬからＨＴＭＬに変換するためのＸＳＬの記述例を示す図である。

【０１４７】

図２１に示されるＸＳＬ４３０において、記述４３１によって、図２０のＸＭＬ４２０の記述４２２で示される<terminal>タグで囲まれた要素に「ｐｄａ」の文字列が含まれているか否か、すなわち、ＸＭＬ４２０はＰＤＡ端末用の処理結果であるか否かが判断される。「ｐｄａ」の文字列が含まれている場合は記述４３２によってＰＤＡ端末用のテンプレートが適用され、「ｐｄａ」の文字列が含まれていない場合は記述４３３によって一般ＰＣ用のテンプレートが適用される。記述４３４にはＰＤＡ端末用のテンプレートが記述され、記述４３５には一般ＰＣ用のテンプレートが記述される。なお、それぞれのテンプレートについては図中では詳述していない。

【０１４８】

例えば、ＰＤＡ端末用のテンプレートを記述する記述４３４では、携帯電話器４２又はＰＤＡ端末４３の表示画面のサイズを考慮して、ＰＣ用では表示されるイメージを表示しない、ＰＣより小さいフォントサイズを指定する、ＰＣより少ない文字数を指定するなどの記述となる。

【０１４９】

ＸＳＬによってＸＭＬをＨＴＭＬに変換する機能をＷｅｂアプリ振分共通ライブラリ１２０が有することによって、一元的に端末種別に応じたＨＴＭＬの作成を行うことができる。よって、Ｗｅｂページ機能３００における各Ｗｅｂアプリケーション３０１は、それぞれに端末種別に対応する処理部を有する必要がない。

【０１５０】

このようにＸＭＬからＨＴＭＬへ変換するＸＳＬを各Ｗｅｂアプリケーション３０１に応じて予め用意しておくことによって、それぞれの処理結果を示すＨＴＭＬを端末４０に提供し、端末４０のＷｅｂブラウザによってその処理結果が表示される。

【０１５１】

端末４０がクライアントＰＣ４１の場合、Ｗｅｂデフォルトアプリ３１０のtopPage()関数の処理結果を示すＨＴＭＬは、例えば、図２２に示されるように記述される。図２２は、ｔｏｐＰａｇｅ．ｃｇｉの処理結果を示すＰＣ用のＨＴＭＬの記述例を示す図である。図２２において、ＰＣ用のＨＴＭＬ４４０は、端末４０がクライアントＰＣ４１の場合に提供され、記述４４１によってＷｅｂデフォルトアプリ３１０のtopPage()関数の処理結果がクライアントＰＣ４１のＷｅｂブラウザに表示される。

【０１５２】

システムステータスを示す記述４４２は、「システムステータス」をフォントサイズ4で表示することを示している。<img src="printer.gif">を示す記述４４３は、イメージを表示することを示している。 システム名：Ｐｒｉｎｔｅｒ　１ を示す記述４４５は、「システム名：Ｐｒｉｎｔｅｒ　１」を表示することを示し、 コメント：ＸＸＸＸＸＸＸＸＸＸ を示す記述４４６は、「コメント：ＸＸＸＸＸＸＸＸＸＸ」を表示することを示し、 システムの状態：正常 を示す記述４４７は、「システムの状態：正常」を表示することを示している。

【０１５３】

このようなＰＣ用のＨＴＭＬ４４０によって、クライアントＰＣ４１のＷｅｂブラウザには、図２３に示すような画面が表示される。図２３は、クライアントＰＣでのシステムステータスの表示例を示す図である。図２３において、ｍａｉｎＦｒａｍｅ．ｃｇｉによって画面４５０が3つのフレームで構成され、各フレームには、ｈｅａｄｅｒＰａｇｅ．ｃｇｉによってヘッダページが表示され、ｍｅｎｕＰａｇｅ．ｃｇｉによってメニューページが表示され、ｔｏｐＰａｇｅ．ｃｇｉによってトップページが表示される。

【０１５４】

トップページには、図２２に示すＨＴＭＬ２４０の記述４４２によって「システムステータス」を示すテキスト４５２が表示され、記述４４４によってイメージ４５４が表示され、記述４４５によって「システム名：Ｐｒｉｎｔｅｒ　1」を示すテキスト４５５が表示され、記述４４６によって「コメント：ＸＸＸＸＸＸＸＸ」を示すテキスト４５６が表示され、記述４４７によって「システムの状態：正常」を示すテキスト４５７が表示される。

【０１５５】

また、ヘッダページを表示するフレームには、例えば、「ＡＡＡＡＡＡ」等のヘッダとしての情報が表示される。メニューページを表示するフレームには、例えば、選択可能なメニュー項目として「システムステータス」、「ネットワーク設定」、「機器構成情報」等のテキスト４５９が表示される。

【０１５６】

「ネットワーク設定」のＨＴＭＬの記述は、例えば、 ネットワーク設定 のように記述される。このＵＲＬには、相対パス情報６０ｂのみが設定されているため、共通パス情報６０ａがそのまま継承され、「ネットワーク設定」に関する情報はＰＣ用にトップページに表示される。つまり、ＷｅｂアプリケーションＩＤを示す「webNetsettingApl」からが相対パスとして設定されており、「../」の記述によって現在の端末種別情報５３ａが継承される。また、「target="top"」の記述によって、「ネットワーク設定」に関する情報はＰＣ用にトップページに表示される。

【０１５７】

「機器構成情報」のＨＴＭＬの記述についても同様に、 機器構成情報 のように記述される。従って、このＵＲＬには、相対パス情報６０ｂのみが設定されているため、共通パス情報６０ａがそのまま継承され、「機器構成情報」に関する情報はＰＣ用にトップページに表示される。

【０１５８】

クライアントＰＣ４１が本発明に係る情報処理装置１００に対して情報を要求した場合、画面４５０のように表示されるが、携帯電話器４２又はＰＤＡ端末４３が同じ情報を要求した場合、図２４に示すようなＰＤＡ端末用のＨＴＭＬが作成され、図２５のような画面がＰＤＡ端末４３に表示される。

【０１５９】

図２４は、ｔｏｐＰａｇｅ．ｃｇｉの処理結果を示すＰＤＡ端末用のＨＴＭＬの記述例を示す図である。図２４において、ＰＤＡ端末用のＨＴＭＬ４４０ａは、端末４０がＰＤＡ端末４３の場合に提供され、記述４４１ａによってＷｅｂデフォルトアプリ３１０のtopPage()関数の処理結果がＰＤＡ端末４３のＷｅｂブラウザに表示される。

【０１６０】

システムステータスを示す記述４４２ａは、「システムステータス」をフォントサイズ３で表示することを示している。記述４４２ａでは、ＰＤＡ端末４３用に文字フォントが小さく設定される。

【０１６１】

ＰＤＡ端末用のＨＴＭＬ４４０ａには、ＰＣ用のＨＴＭＬ４４０の記述４４３に相当する記述が存在しない。

【０１６２】

ＰＣ用のＨＴＭＬ４４０の記述４４５及び４４６と同様に、 システム名：Ｐｒｉｎｔｅｒ　１ を示す記述４４５ａは、「システム名：Ｐｒｉｎｔｅｒ１」を表示することを示し、 コメント：ＸＸＸＸＸＸＸＸＸＸ を示す記述４４６ａは、「コメント：ＸＸＸＸＸＸＸＸＸＸ」を表示することを示す。

【０１６３】

クライアントＰＣ４１に比べて表示領域の小さいＰＤＡ端末４３では、ＰＣ用のＨＴＭＬ４４０の記述４４７に相当する記述４４７ａにおいて、「システムの状態：正常」を表示する代わりに、例えば、「状態：正常」を表示することを示し、表示文字数を少なくして表示する。

【０１６４】

また、ＰＤＡ端末４３では画面をフレームによって複数に分割することが適切

でないため、クライアントＰＣ４１では他のフレームに表示される情報を、ＰＤＡ端末４３では１つの画面（ページ）内に表示する必要がある。記述４４８ａは、ＰＣ用ではメニューページを表示するＨＴＭＬに記述されるものであって、ＰＤＡ端末用のＨＴＭＬ４４０ａでは、 ネットワーク設定 及び 機器構成情報 のように記述し、クライアントＰＣ４１のメニューページに表示される選択可能な項目のうち「ネットワーク設定」と「機器構成情報」とを表示することを示す。記述４４８ａにおいて、ＰＣ用のＨＴＭＬ４４０では、表示すべきフレームをtarget=”top”によって指定していたが、その記述が削除される。

【０１６５】

このようなＰＤＡ端末用のＨＴＭＬ４４０ａによって、ＰＤＡ端末４３のＷｅｂブラウザには、図２５に示すような画面が表示される。図２５は、ＰＤＡでのシステムステータスの表示例を示す図である。図２５において、画面４５０ａは、フレームによって分割されない１つの画面で構成される。

【０１６６】

画面４５０ａには、図２４に示すＨＴＭＬ２４０ａの記述４４２ａによって「システムステータス」を示すテキスト４５２ａが表示され、クライアントＰＣ４１には表示されるイメージを表示することなく、記述４４５ａによって「システム名：Ｐｒｉｎｔｅｒ　１」を示すテキスト４５５が表示される。記述４４６ａによって「コメント：ＸＸＸＸＸＸＸＸＸＸ」を示すテキスト４５６ａが表示され、記述４４７ａによってクライアントＰＣ４１より文字数の少ない「状態：正常」を示すテキスト４５７ａが表示される。

【０１６７】

更に、クライアントＰＣ４１ではメニューページに表示される選択可能な「ネットワーク設定」及び「機器構成情報」が、記述４４８ａによって選択可能な「ネットワーク設定」及び「機器構成情報」を示すテキスト４５８ａとして表示される。

【０１６８】

画面４５０ａが携帯端末４２又はＰＤＡ端末４３に表示された状態で、ＰＤＡ端末４３のユーザが「ネットワーク設定」を選択すると、共通パス情報６０ａが継承され相対パス情報６０ｂのみが「ネットワーク設定」を示すＵＲＬに変更されるため、図２６に示すような画面が携帯端末４２又はＰＤＡ端末４３に表示される。図２６は、ＰＤＡ端末でのネットワーク設定の表示例を示す図である。

【０１６９】

図２６において、画面４６０ａは、画面４５０ａからページ遷移して表示された画面であって、ページ遷移時に共通パス情報６０ａが継承されるため、携帯端末４２又はＰＤＡ端末４３の表示画面のサイズに応じて「ネットワーク設定」のページが表示される。携帯端末４２又はＰＤＡ端末４３の利用者は、表示画面のサイズに合せて「ネットワーク設定」のページを表示させるための設定を何ら行なわない。

【０１７０】

従来のような端末４０の端末種別が識別されない情報処理装置に対して、ページ遷移を行うと、クライアントＰＣ４１への表示画面のサイズに応じた表示が成されるため、図２７に示すような情報提供となってしまう。端末種別情報５３ａが継承されないため、大き過ぎる文字で表示されていまい、携帯端末４２又はＰＤＡ端末４３の表示画面に適した情報を提供することができない。

【０１７１】

本願発明によれば、ＨＴＴＰデーモン２が、端末４０からの最初の要求を受信した時にｄｅｆａｕｌｔ．ｃｇｉを呼び出すようにＵＲＬを構成するため、ｄｅｆａｕｌｔ．ｃｇｉによって共通パス情報６０ａを持ったＵＲＬを情報処理装置１００にて自動的に設定することができる。従って、端末４０の利用者は、表示画面を意識した情報の入力を行う必要がない。

【０１７２】

共通パス情報６０ａを相対パス情報６０ｂの前に構成するため、ページからのリンクは相対パスを指定するのみで良く、ＨＴＭＬの作成を容易とすることができる。

【０１７３】

また、情報処理装置１００は、スタイルシートを用いて、共通パス情報６０ａの端末種別情報５３ａに基づいて、ＸＭＬからＨＴＭＬへの変換を行う。従って、情報処理装置１００において、Ｗｅｂアプリケーション３０１の開発者は、端末種別を意識したアプリケーション開発をする必要がなく、開発者の負担を軽減することができる。つまり、各Ｗｅｂアプリケーション３０１は、端末種別の違いに依存せずに、統一した処理結果を出力することができる。例えば、表示画面のサイズの違い、フレームによる表示画面の分割等が、Ｗｅｂアプリケーション３０１の開発に影響を与えることがない。

【０１７４】

上記実施例において、端末種別として、クライアントＰＣ４１を「ＰＣ」、携帯電話器４２及びＰＤＡ端末４３を「ＰＤＡ」として分類したが、表示画面に応じた分類が可能であれば良いため、「ＰＣ」、「ＰＤＡ」の他に更に多くの端末種別があっても良い。また、分類は表示画面の特性に応じた分類であればよい。

【０１７５】

また、本願発明は、端末４０として巨大スクリーンが情報処理装置１００に接続される場合にも適応することができる。つまり、巨大スクリーンに応じて、文字を大きく変更したり、文字数を多くして情報量を多くしたりすることが可能となる。また、より大きいイメージ又は複数のイメージを表示して、効果的に情報を提供することができる。

【０１７６】

このような端末４０に応じた情報を提供するために、本発明に係る情報処理装置１００では、各スタイルシート４３０の端末種別に応じたテンプレートにて、Ｗｅｂページ機能３００から提供される処理結果の表示形式を調整及びイメージ等の付加を行なっている。

【０１７７】

【発明の効果】

以上、説明してきたように、本願発明によれば、端末からの要求時に端末種別が共通パス情報としてＵＲＬの一部に設定されるため、複数のＷｅｂアプリケーション間でページ遷移した場合においても、端末のＷｅｂブラウザに常に同一の

端末種別に対応した情報提供を行うことができる。また、処理結果のＸＭＬによる記述と、端末種別を判断してＸＭＬからＨＴＭＬへ変換する処理とを、複数のＷｅｂアプリケーションから切り離して構成することができる。

【図面の簡単な説明】

【図１】

本発明の一実施例に係る情報処理装置のハードウェア構成を示すブロック図である。

【図２】

情報処理装置の機能構成を示すブロック図である。

【図３】

ＵＲＬ構成を示す図である。

【図４】

Ｗｅｂアプリケーションの実行を説明するためのブロック図である。

【図５】

図５は、ハッシュリストを示す図である。

【図６】

ｄｅｆａｕｌｔ．ｃｇｉによる処理フローを示す図である。

【図７】

ｄｅｆａｕｌｔ．ｃｇｉによる処理フローを示す図である。

【図８】

端末情報の解析処理について説明するためのフローチャート図である。

【図９】

ＨＴＴＰ接続情報のデータ構造を示す図である。

【図１０】

ＨＴＴＰ要求情報のデータ構造を示す図である。

【図１１】

ヘッダ情報リストを示す図である。

【図１２】

Ｗｅｂブラウザ種別―端末種別対応表を示す図である。

【図１３】

ＷｅｂブラウザＯＳ－端末種別対応表を示す図である。

【図１４】

デフォルトＨＴＭＬ出力の例を示す図である。

【図１５】

ｍａｉｎＦｒａｍｅ．ｃｇｉによる処理フローを示す図である。

【図１６】

ｍａｉｎＦｒａｍｅ．ｃｇｉによる処理フローを示す図である。

【図１７】

ＰＣ用のメインフレームＨＴＭＬ出力の例を示す図である。

【図１８】

ｔｏｐＰａｇｅ．ｃｇｉによる処理フローを示す図である。

【図１９】

ｔｏｐＰａｇｅ．ｃｇｉによる処理フローを示す図である。

【図２０】

ＸＭＬで記述されたｔｏｐＰａｇｅ．ｃｇｉによる処理結果の例を示す図である。

【図２１】

処理結果をＸＭＬからＨＴＭＬに変換するためのＸＳＬの記述例を示す図である。

【図２２】

ｔｏｐＰａｇｅ．ｃｇｉの処理結果を示すＰＣ用のＨＴＭＬの記述例を示す図である。

【図２３】

クライアントＰＣでのシステムステータスの表示例を示す図である。

【図２４】

ｔｏｐＰａｇｅ．ｃｇｉの処理結果を示すＰＤＡ端末用のＨＴＭＬの記述例を示す図である。

【図２５】

ＰＤＡ端末でのシステムステータスの表示例を示す図である。

【図２６】

ＰＤＡ端末でのネットワーク設定の表示例を示す図である。

【図２７】

一般ＰＣ用のページをＰＤＡ端末にそのまま表示した例を示す図である。

【符号の説明】

２　　ＨＴＴＰデーモン

１６　　インターネット

１７　　ＡＰＩ

４０　　端末

４１　　クライアントＰＣ

４２　　携帯電話器

４３　　ＰＤＡ

１００　　情報処理装置

１０１　　ネットワーク制御部

１０３　　プリンタ制御部

１０５　　スキャナ制御部

１１０　　シーケンス制御ライブラリ

１２０　　Ｗｅｂアプリ振分共通ライブラリ

２００　　Ｗｅｂページハンドラ

２０１　　ＳＯＡＰライブラリ

２０３　　ＸＭＬライブラリ

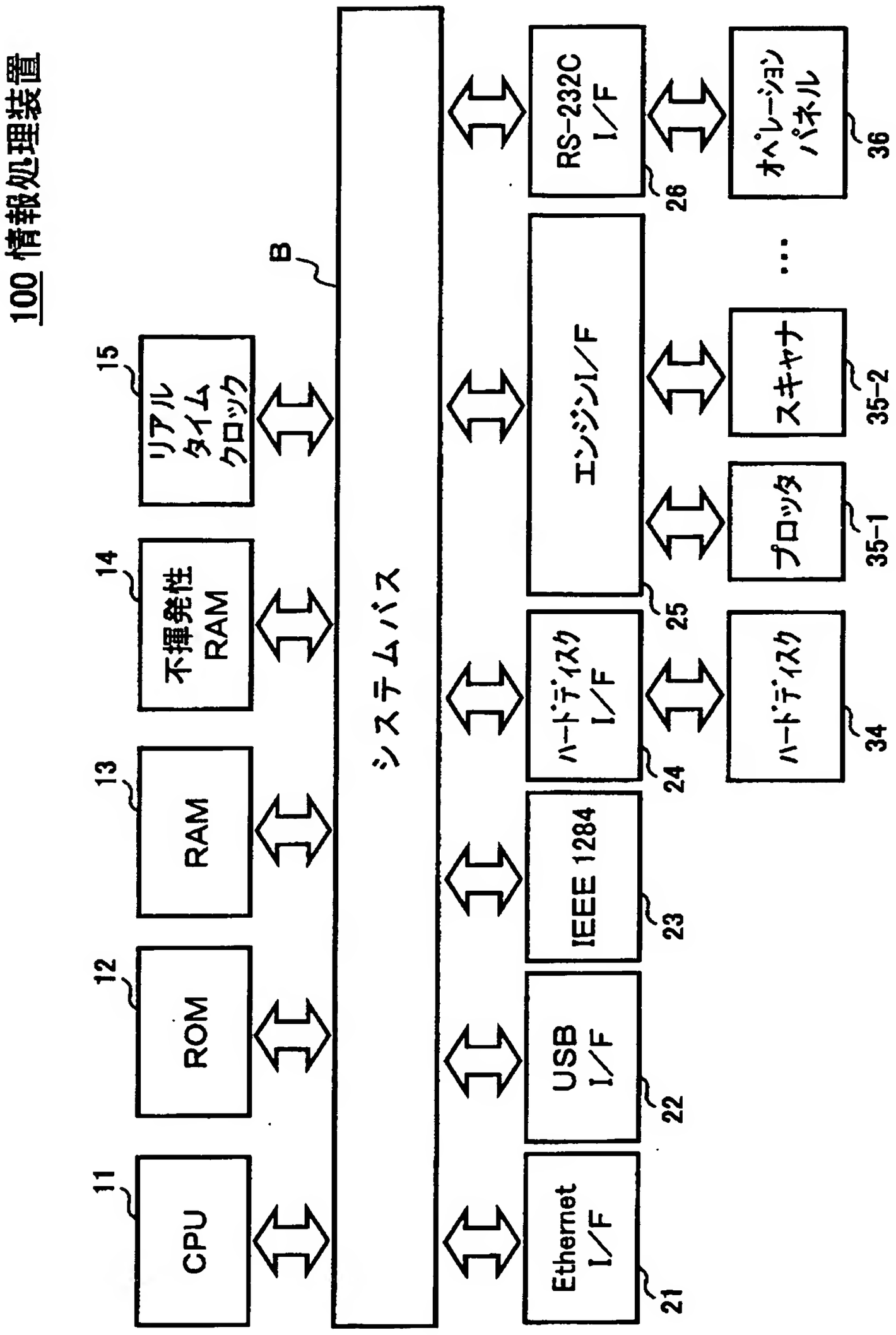
２０５　　ＸＳＬＴプロセッサ

３００　　Ｗｅｂページ機能（ＷＰＦ）

【書類名】　図面

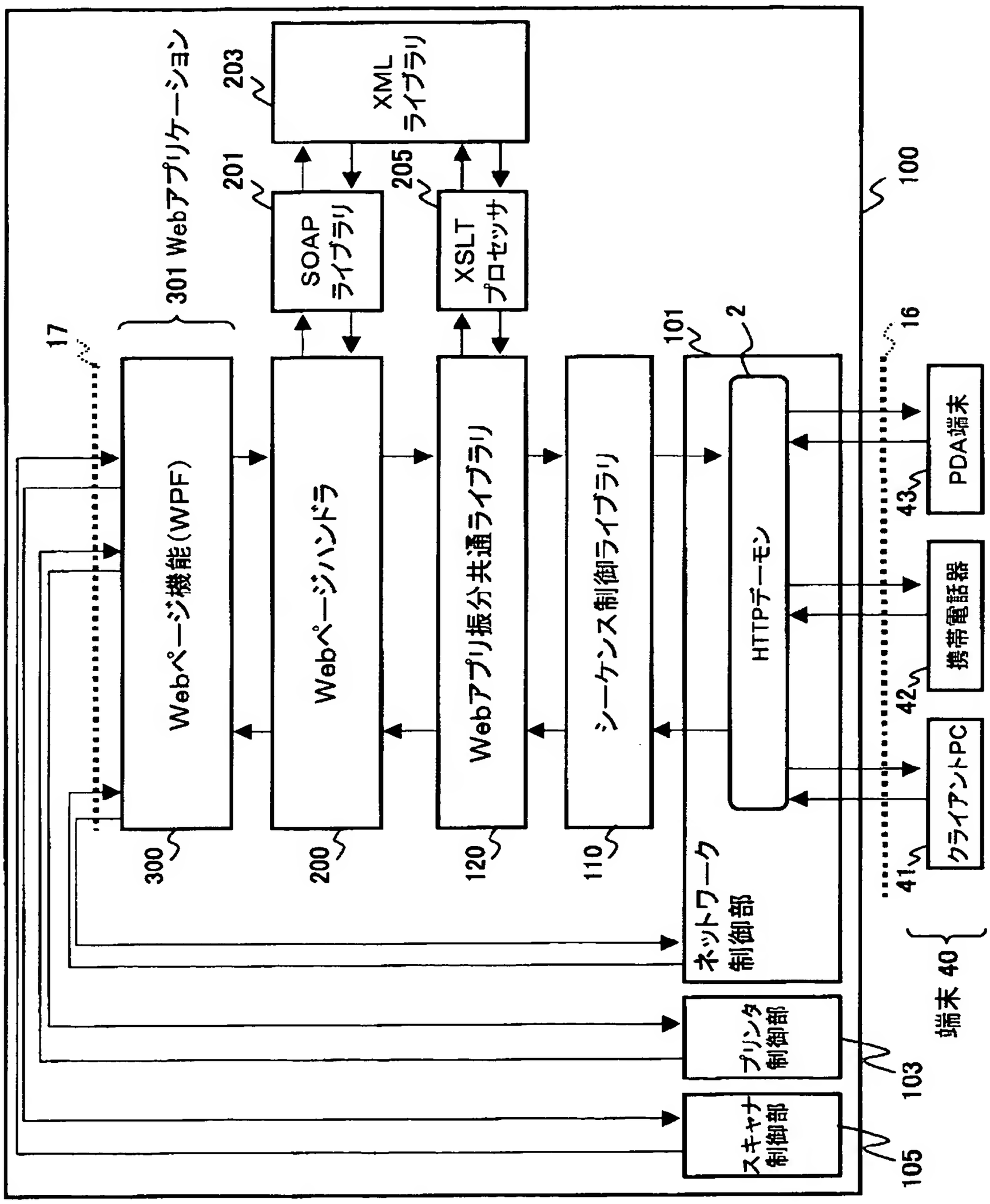
【図１】

本発明の一実施例に係る情報処理装置のハードウェア構成を示すブロック図



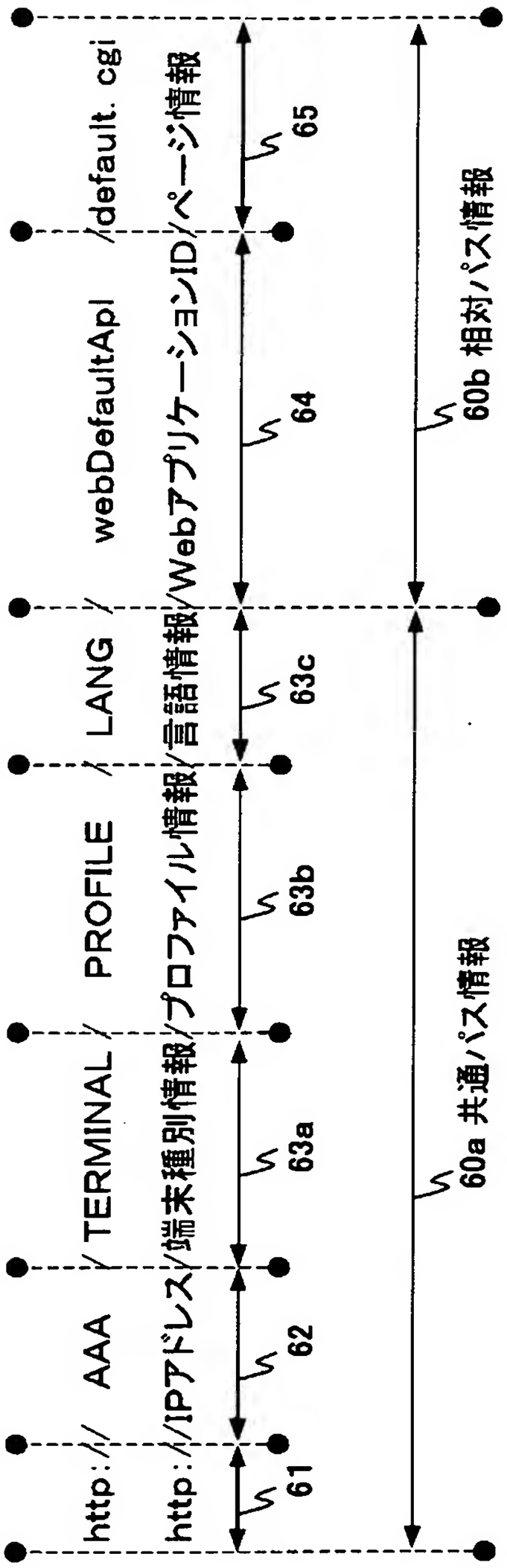
【図２】

情報処理装置の機能構成を示すブロック図



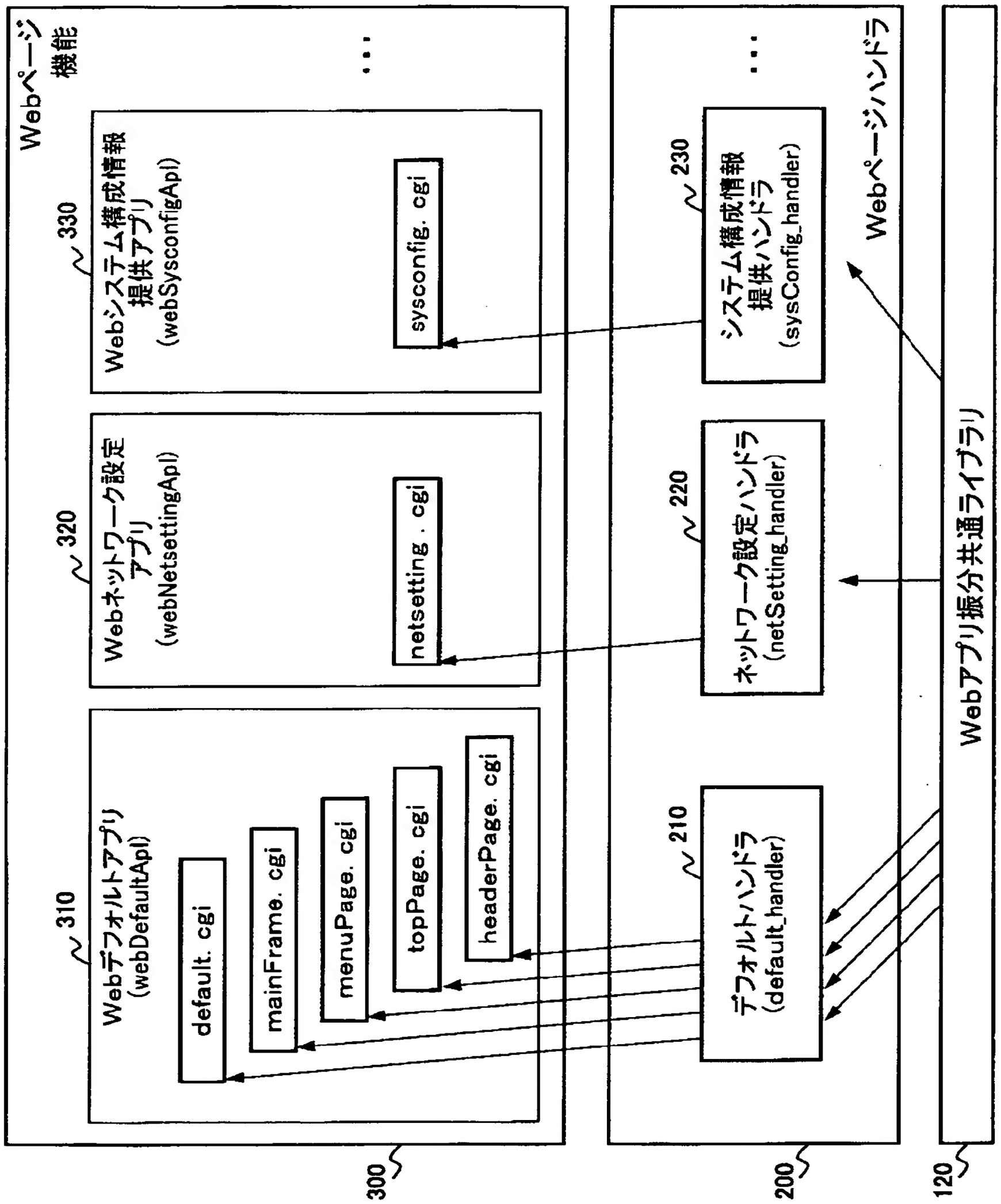
【図３】

URL構成を示す図



【図４】

Webアプリケーションの実行を説明するためのブロック図



【図５】

ハッシュリストを示す図

(A)

50a

WebアプリケーションID	Webアプリケーションを呼ぶための Webページハンドラ
webDefaultApl	default__handler
webNetsettingApl	netSetting__handler
webSysconfigApl	sysConfig__handler
...	...

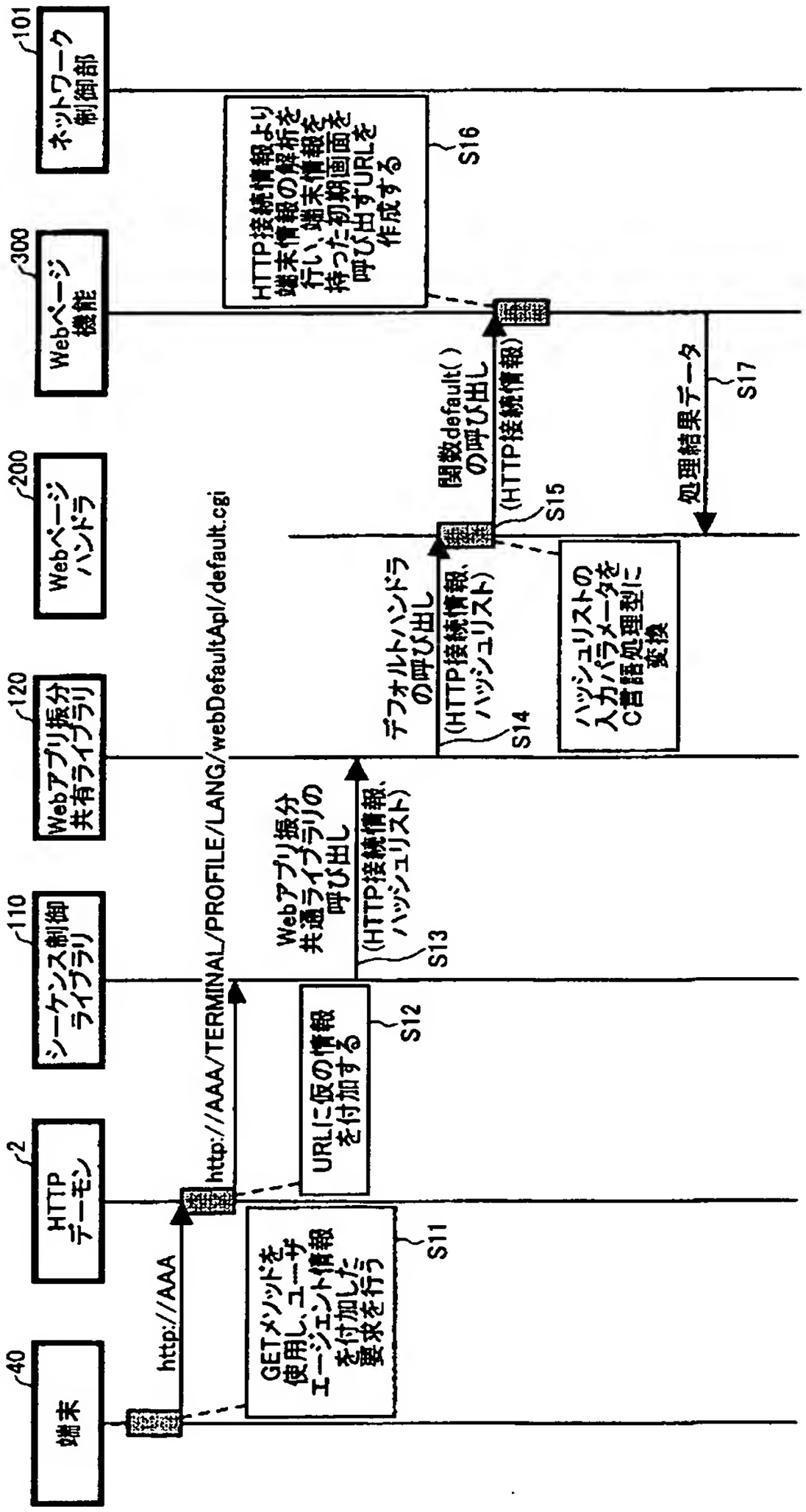
(B)

50b

Webページ機能ID	関数	スタイルシート
default. cgi	default()	default. xsl
mainFrame. cgi	mainFrame()	mainFrame. xsl
menuPage. cgi	menuPage()	menuPage. xsl
topPage. cgi	topPage()	topPage. xsl
headerPage. cgi	headerPage()	headerPage. xsl
netsetting. cgi	netsetting()	netsetting. xsl
sysconfig. cgi	sysconfig()	sysconfig. xsl
...	...	...

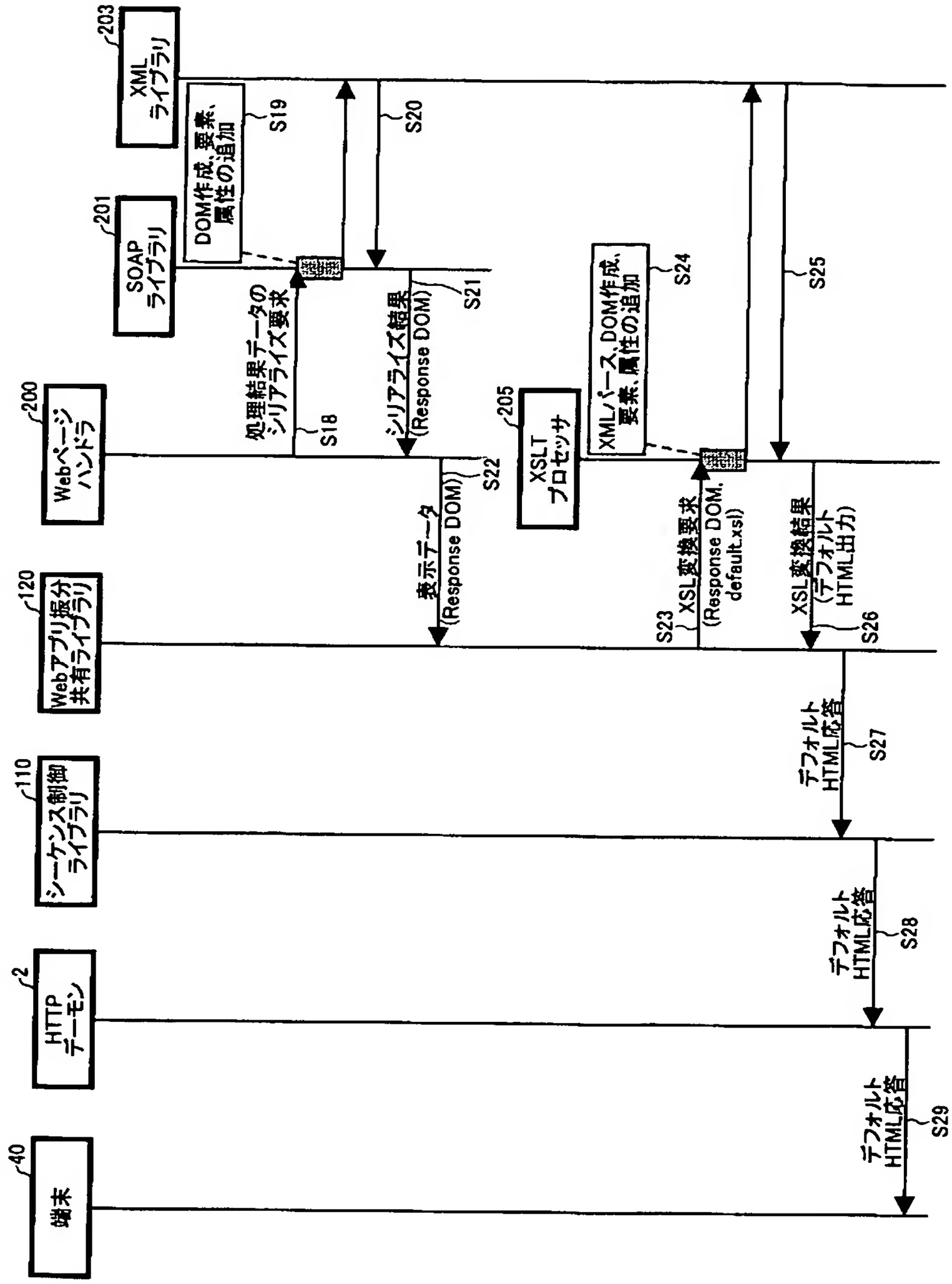
【図６】

default. cgiによる処理フローを示す図

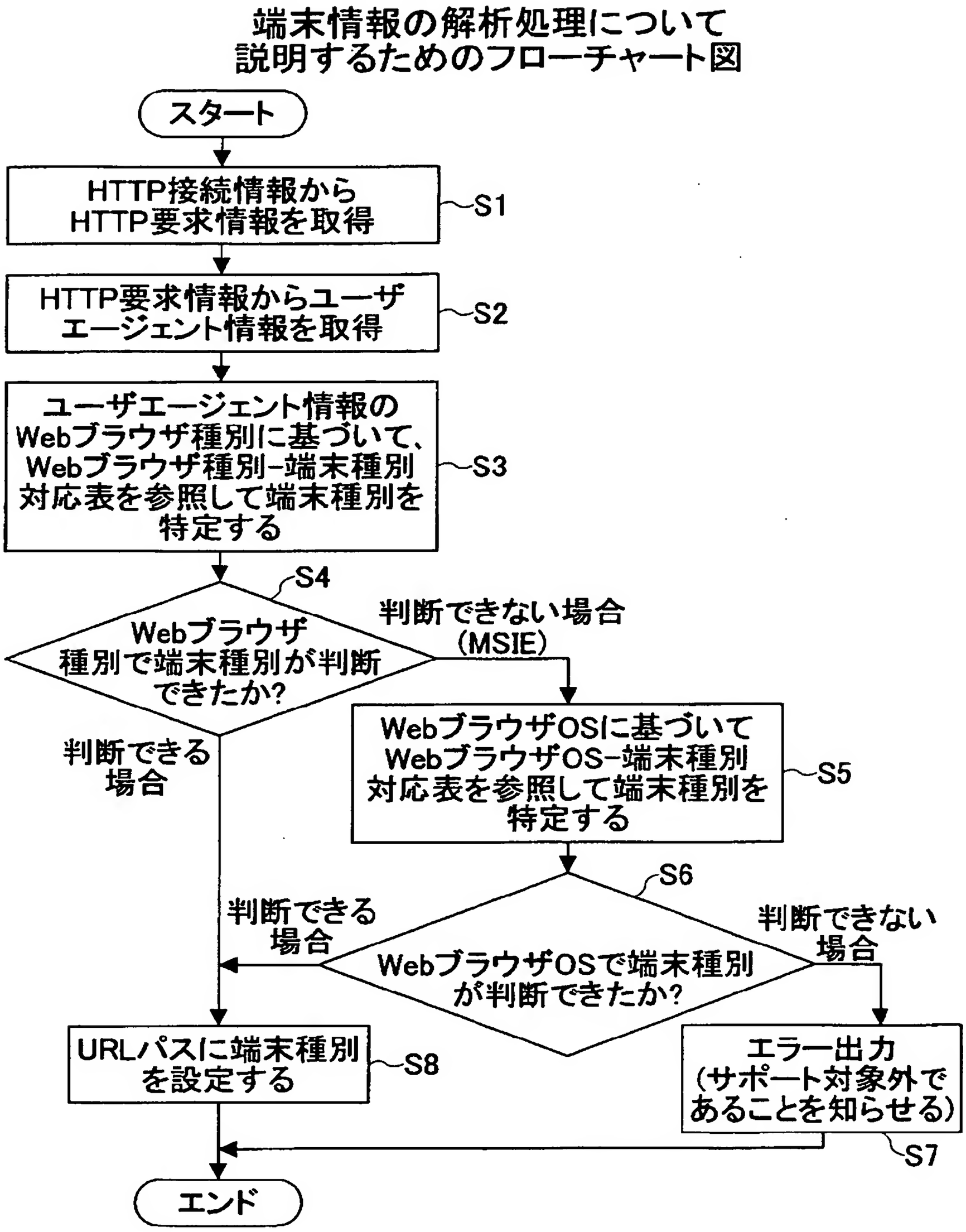


【図７】

default. cgilによる処理フローを示す図

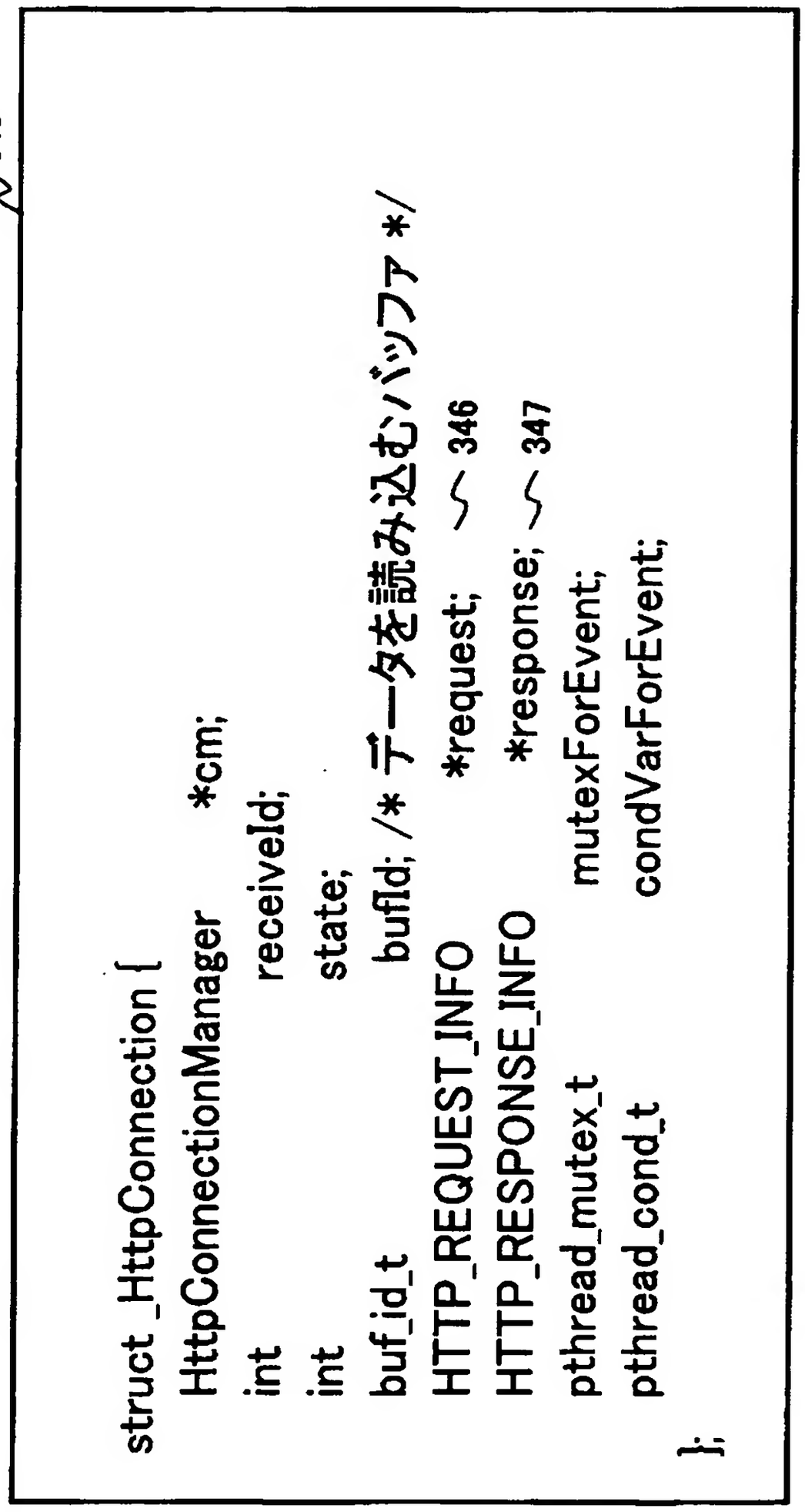


【図８】


端末情報の解析処理について
説明するためのフローチャート図
スタート
HTTP接続情報から
HTTP要求情報を取得
S1
HTTP要求情報からユーザ
エージェント情報を取得
S2
ユーザエージェント情報の
Webブラウザ種別に基づいて、
Webブラウザ種別−端末種別
対応表を参照して端末種別を
特定する
S3
S4
Webブラウザ
種別で端末種別が判断
できたか?
判断できない場合
(MSIE)
判断できる
場合
WebブラウザOSに基づいて
WebブラウザOS−端末種別
対応表を参照して端末種別を
特定する
S5
S6
WebブラウザOSで端末種別
が判断できたか?
判断できる
場合
判断できない
場合
URLパスに端末種別
を設定する
S8
エラー出力
(サポート対象外で
あることを知らせる)
S7
エンド

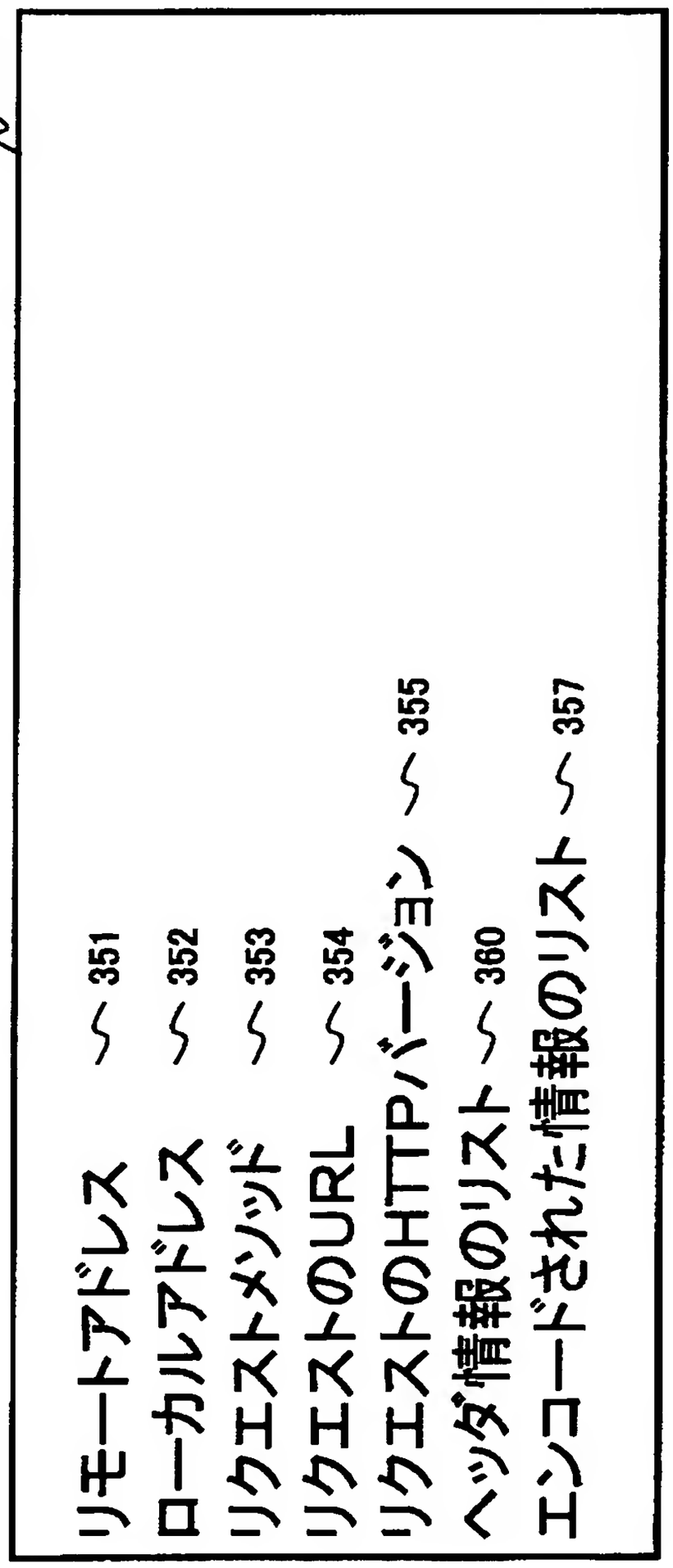
【図９】

HTTP接続情報のデータ構造を示す図



【図１０】

HTTP要求情報のデータ構造を示す図



【図１１】

ヘッダ情報リストを示す図

360

GET HTTP/1.0
Accept:image/gif,image/xbitmap,image/ipeg,image/pipeg,application/vnd.spowerpoint,application/
Voln.ms-excel,application/msword,application/pdf,*/*
accept-Language:en-us;ge;q=0.9;ja;q=0.7;it;q=0.6;sp;q=0.4;pt;q=0.3
User-Agent:Mozilla/4.0{compatible;MSIE 6.0;Windows NT 5.0;Q312461;NET CLR 1.0.3705 365
Host:133.139.49.79
Cookie:cookieOnOffCheck=on
Connection:keep-alive
Accept-encouding.gzip,deflate

【図１２】

Ｗｅｂブラウザ種別―端末種別対応表を示す図

370

Webブラウザ種別	端末種別
DOCOMO	IMODE
xiino	PDA
MSPIE	PDA
HandHTTP	PDA
MSIE	O
Mozilla	PC
mozilla/5	PC
Netscape6	PC

【図１３】

ＷｅｂブラウザＯＳ―端末種別応表を示す図

380

WebブラウザOS	端末種別
Windows3.1	PC
Windows95	PC
Windows98	PC
WindowsME	PC
WindowsNT	PC
Windows2000	PC
WindowsXP	PC
Mac 68k	PC
Mac PowerPC	PC
Solaris	PC
Linux	PC
FreeBSD	PC
AIX	PC
IRIX	PC
HP	PC
OS/2	PC
WindowsCE	PDA

【図１４】

デフォルトHTML出力の例を示す図

401

403

400

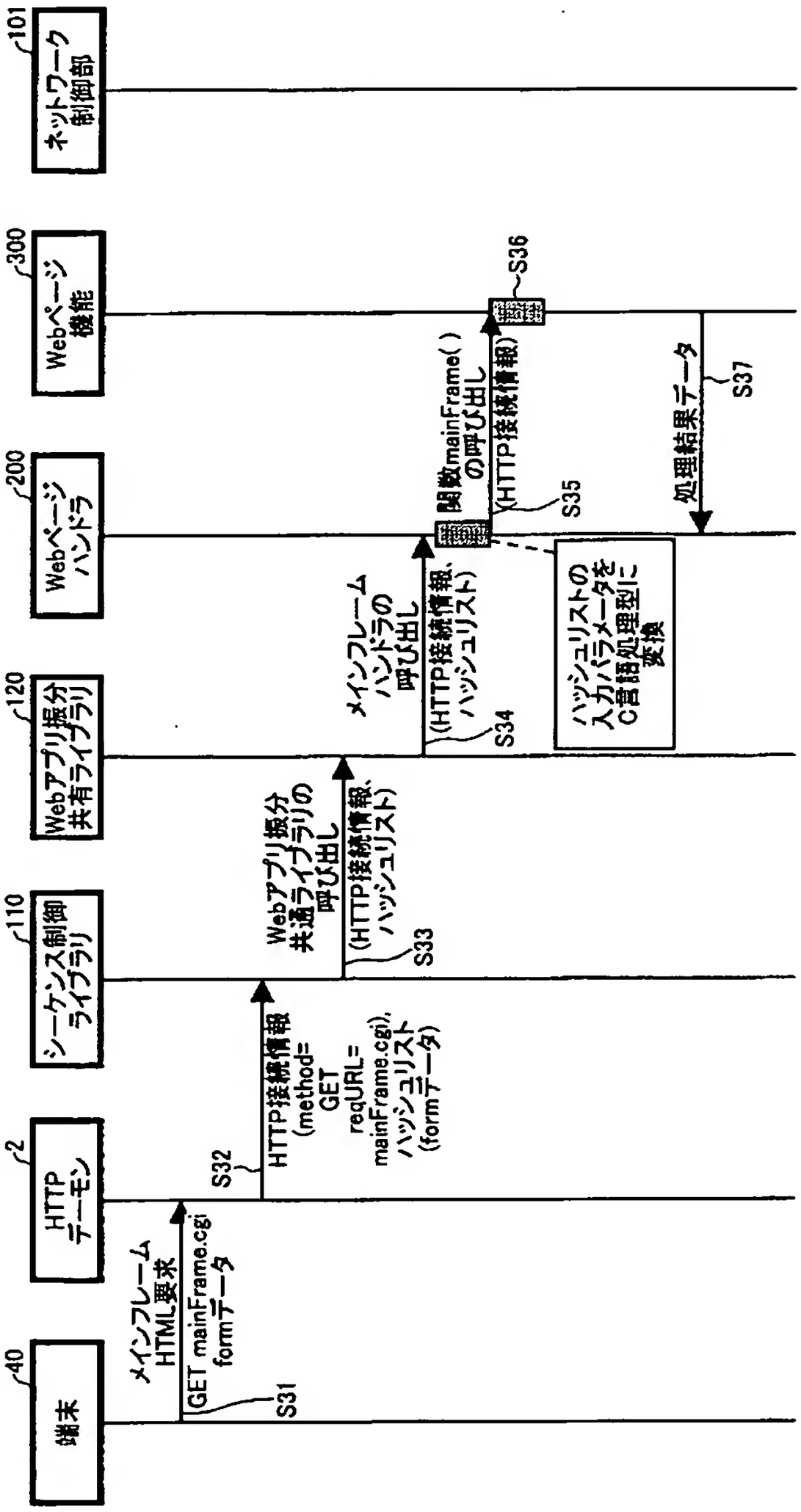
402

```
<html>
<head>
<title>default page</title>
<script language="javascript"
<!--
function jumpToMainFrame(
self.document.cookie="cookie
self.location.pathname="/pc/a
}
// -->
</script>
</head>
<body onLoad="jumpToMain
</body>
</html>
```

```
pt">
ne(){
kieOnOffChecker=on; path=/";
c/admin/ja/webDefaultApl/mainFrame.cgi";
inFrame()">
```

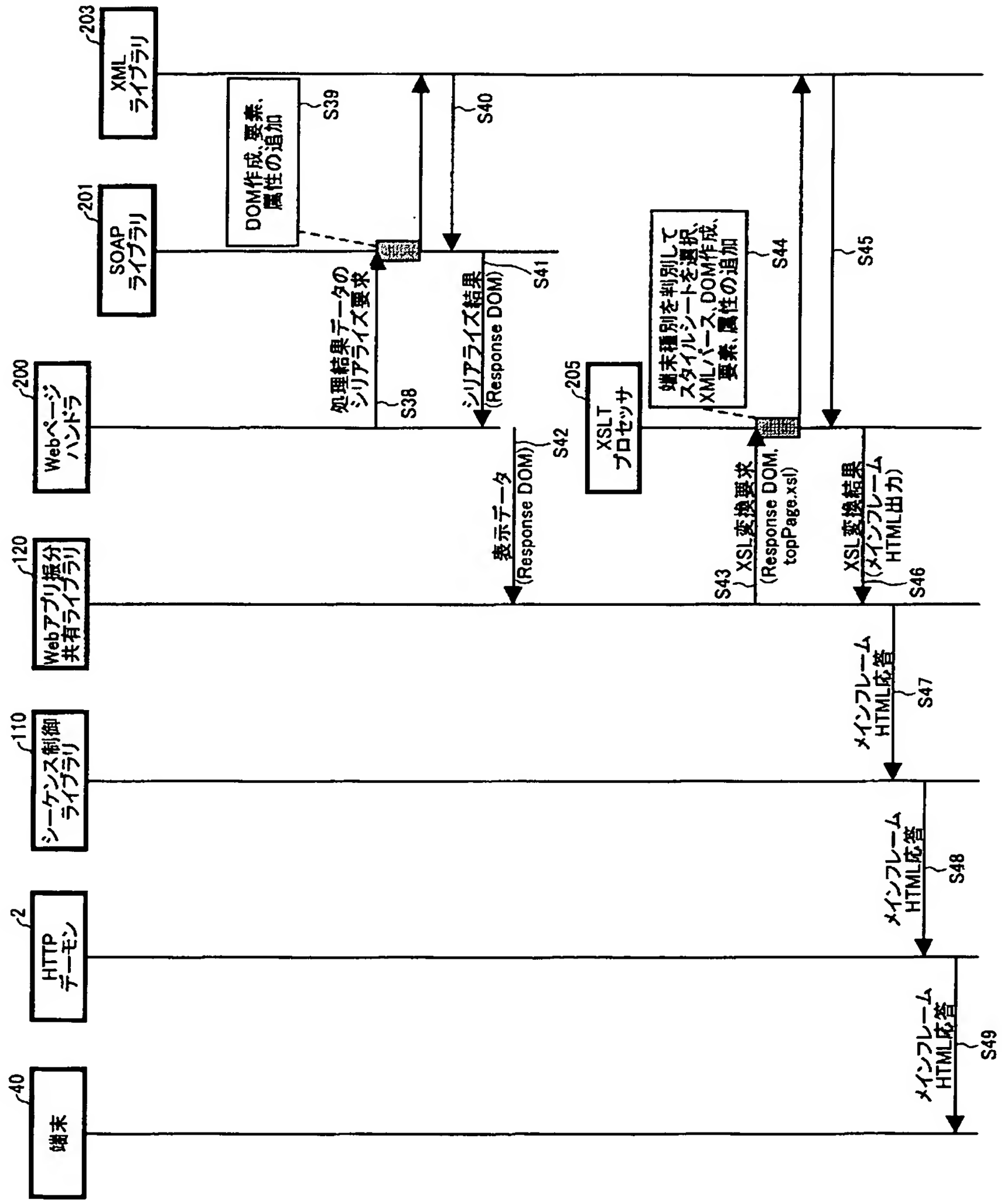
【図１５】

mainFrame. cgiによる処理フローを示す図



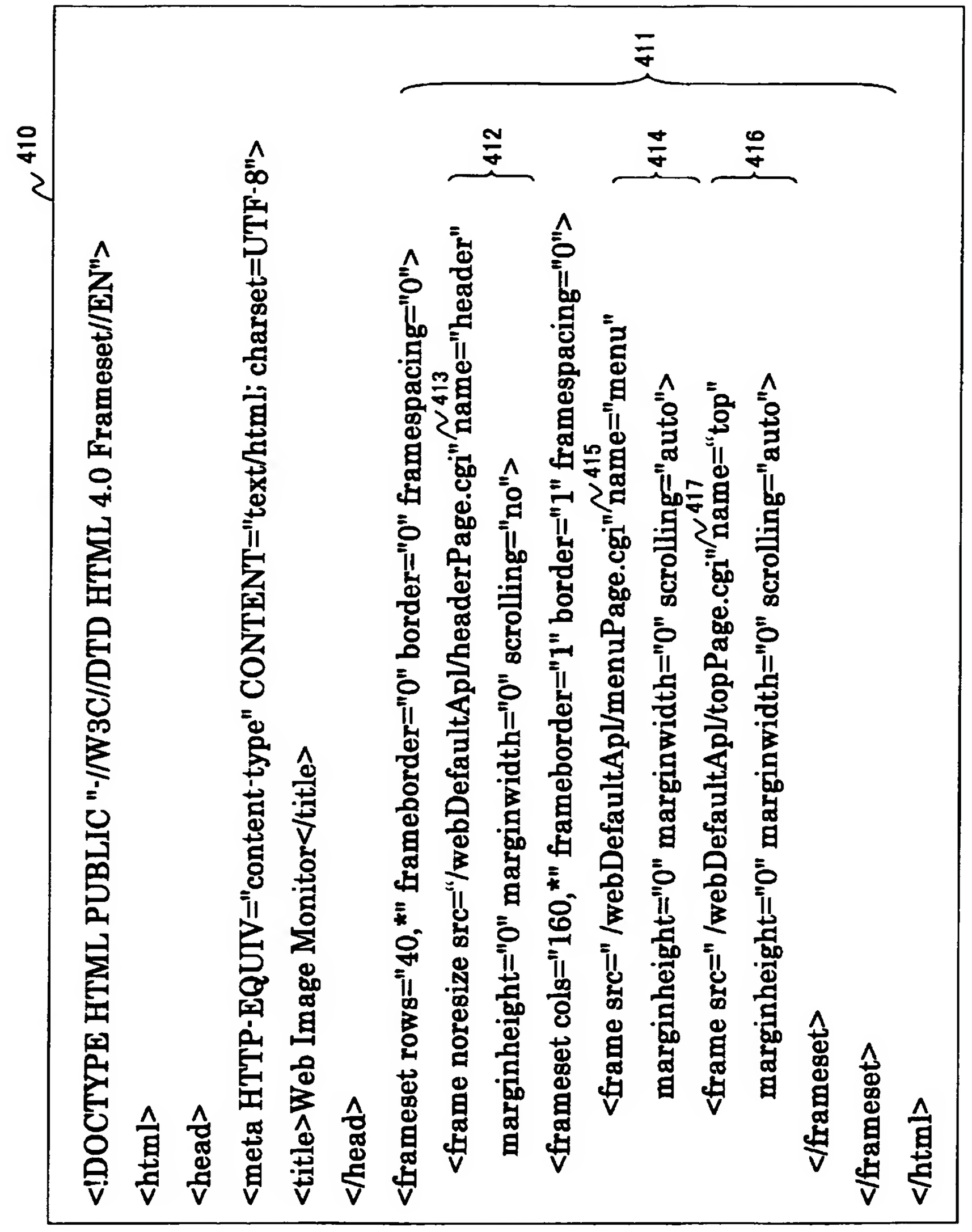
【図１６】

mainFrame. cgiによる処理フローを示す図



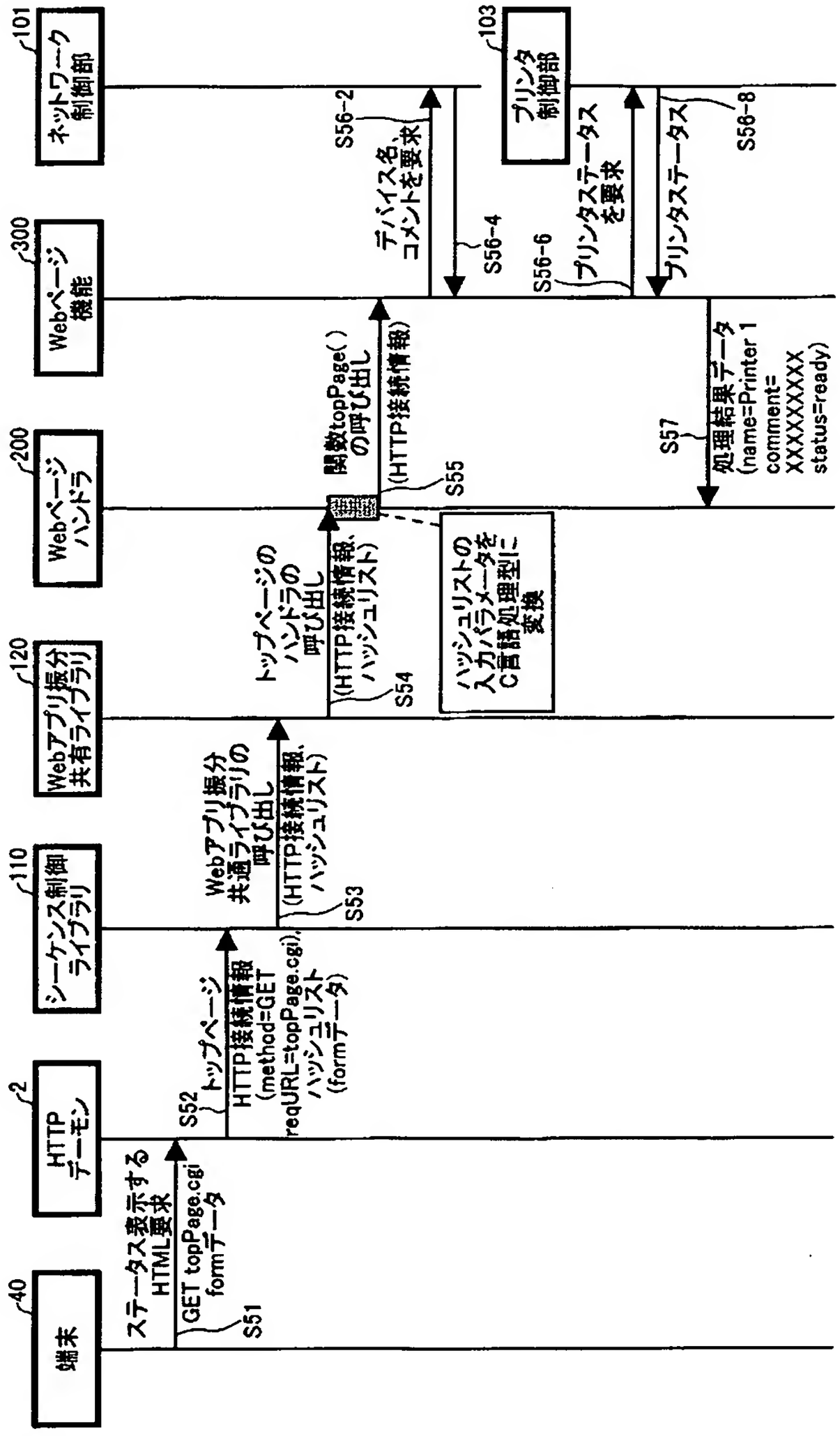
【図１７】

PC用のメインフレームHTML出力の例を示す図



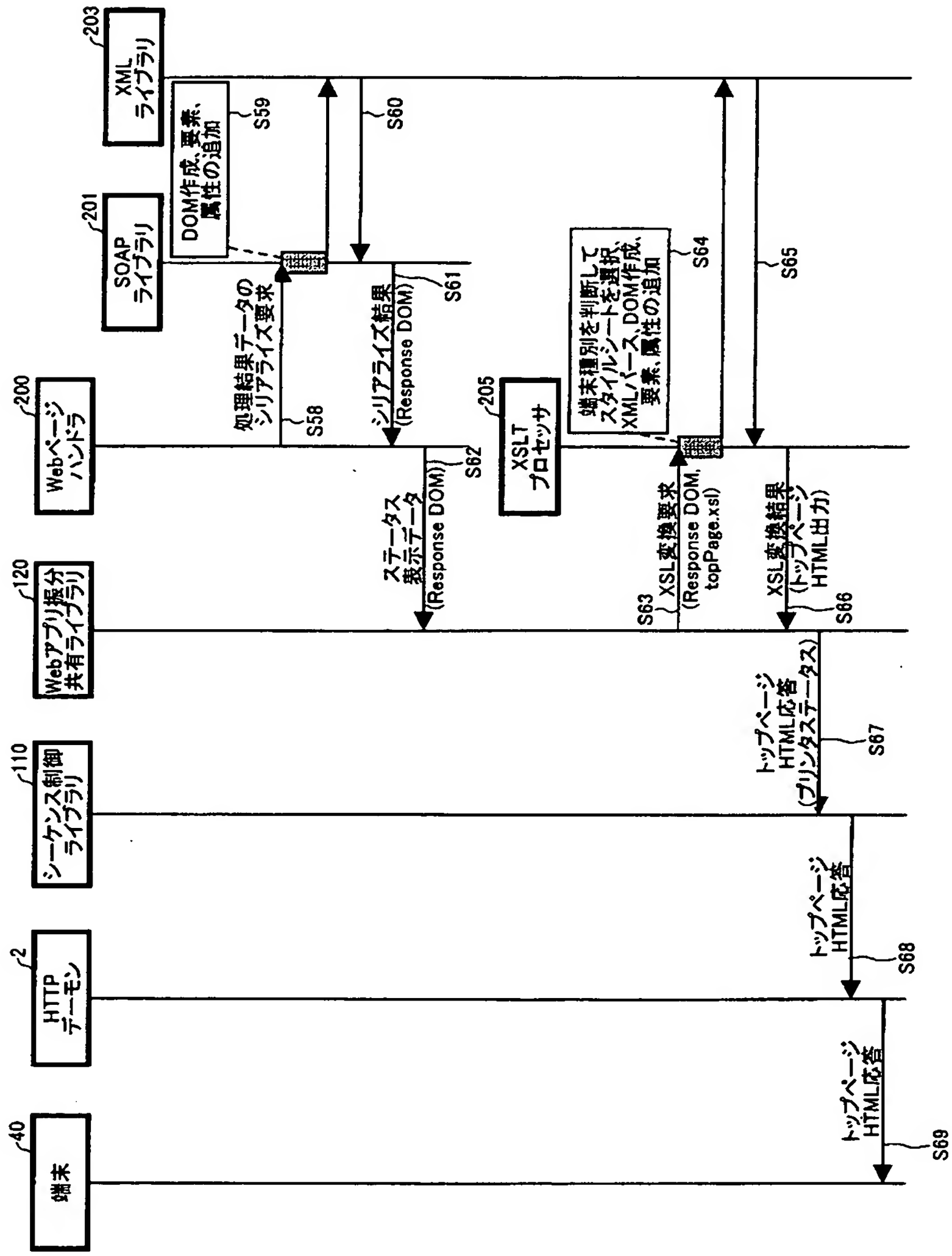
【図１８】

topPage. cgiによる処理フローを示す図



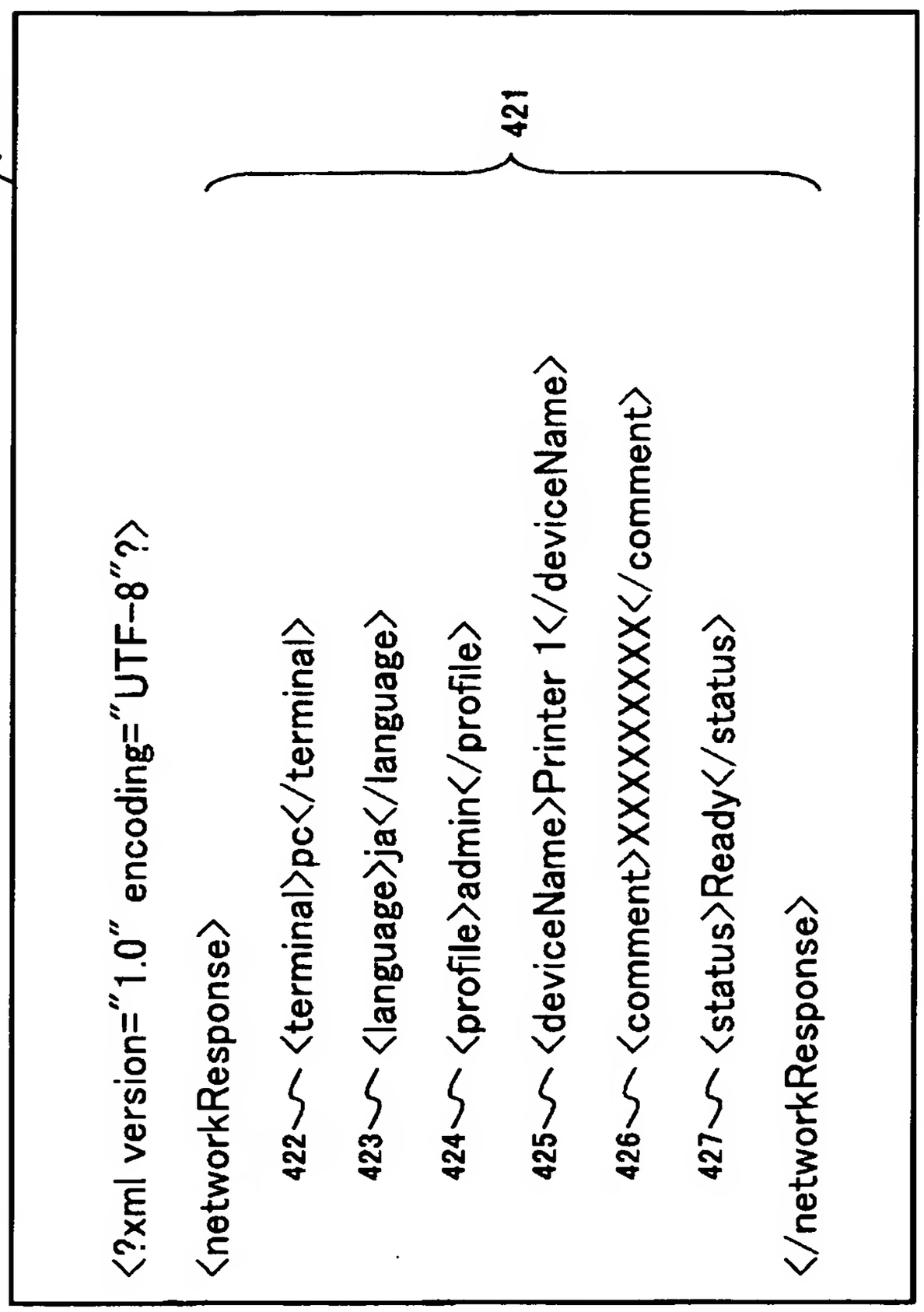
【図１９】

topPage. cgiによる処理フローを示す図



【図２０】

XMLで記述されたtopPage.cgiによる
処理結果の例を示す図



【図２１】

処理結果をＸＭＬからＨＴＭＬに変換するための
ＸＳＬの記述例を示す図

430

```
<?xml version="1.0" encoding="UTF-8" ?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform" xmlns:sf="http://www.rr
 rr.co.jp/xmlns/xslt/rdh/common" version="1.0">
  <xsl:output method="html" encoding="UTF-8" />
<xsl:template match="/">
<html>
        <body>
                <xsl:apply-templates />
        </body>
</html>
</xsl:template>

<!-- Templates for body -->
<xsl:template match="networkResponse">
<!-- Display the page title -->
<xsl:call-template name="statusTitle4CommonProfile" />
<!-- Check current terminal type(pda or pc) and call suitable template -->
        <xsl:choose>
                <xsl:when test="contains(//urldevice, 'pda')">  ～431
                        <xsl:call-template name="networkResponse_pda" /> ～432
                </xsl:when>
                <xsl:otherwise>
                        <xsl:call-template name="networkResponse" /> ～433
                </xsl:otherwise>
        </xsl:choose>
</xsl:template>

<!-- 共通項目 -->
<xsl:template name="networkTitle4CommonProfile">
        <!-- 略 -->
</xsl:template>

<!-- PDA用 スタイルシート -->
<xsl:template name="networkResponse_pda">
        <!-- 略 -->                                  }434
</xsl:template>

<!-- PC用 スタイルシート -->
<xsl:template name="networkResponse">
        <!-- 略 -->                                  }435
</xsl:template>
</xsl:stylesheet>
```

【図２２】

topPage.cgiの処理結果を示すPC用のHTMLの記述例を示す図

```
<html>

<head>
<meta http-equiv="Content-Language" content="ja">
<meta http-equiv="Content-Type" content="text/html; charset=shift_jis">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>システムステータス</title>
</head>

<body>
<p><b><font size="4">システムステータス</font></b></p>
<p>システム名：Printer  1</p>
<p>コメント：XXXXXXXXX</p>
<p>システムの状態：正常</p>
<p></p>

</body>

</html>
```

440

443

442

444

445

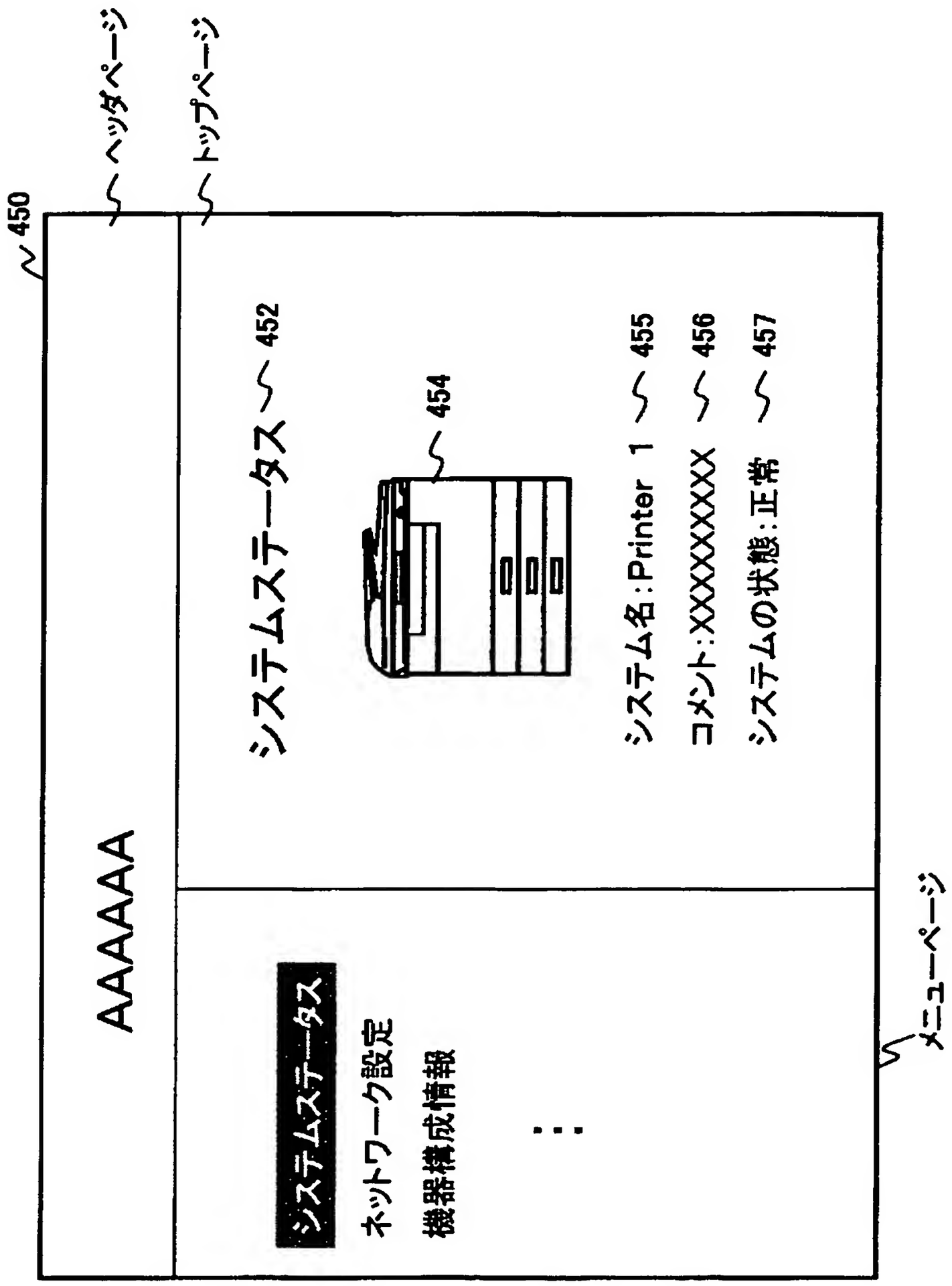
446

447

441

【図２３】

クライアントPCでのシステムステータスの表示例を示す図



【図２４】

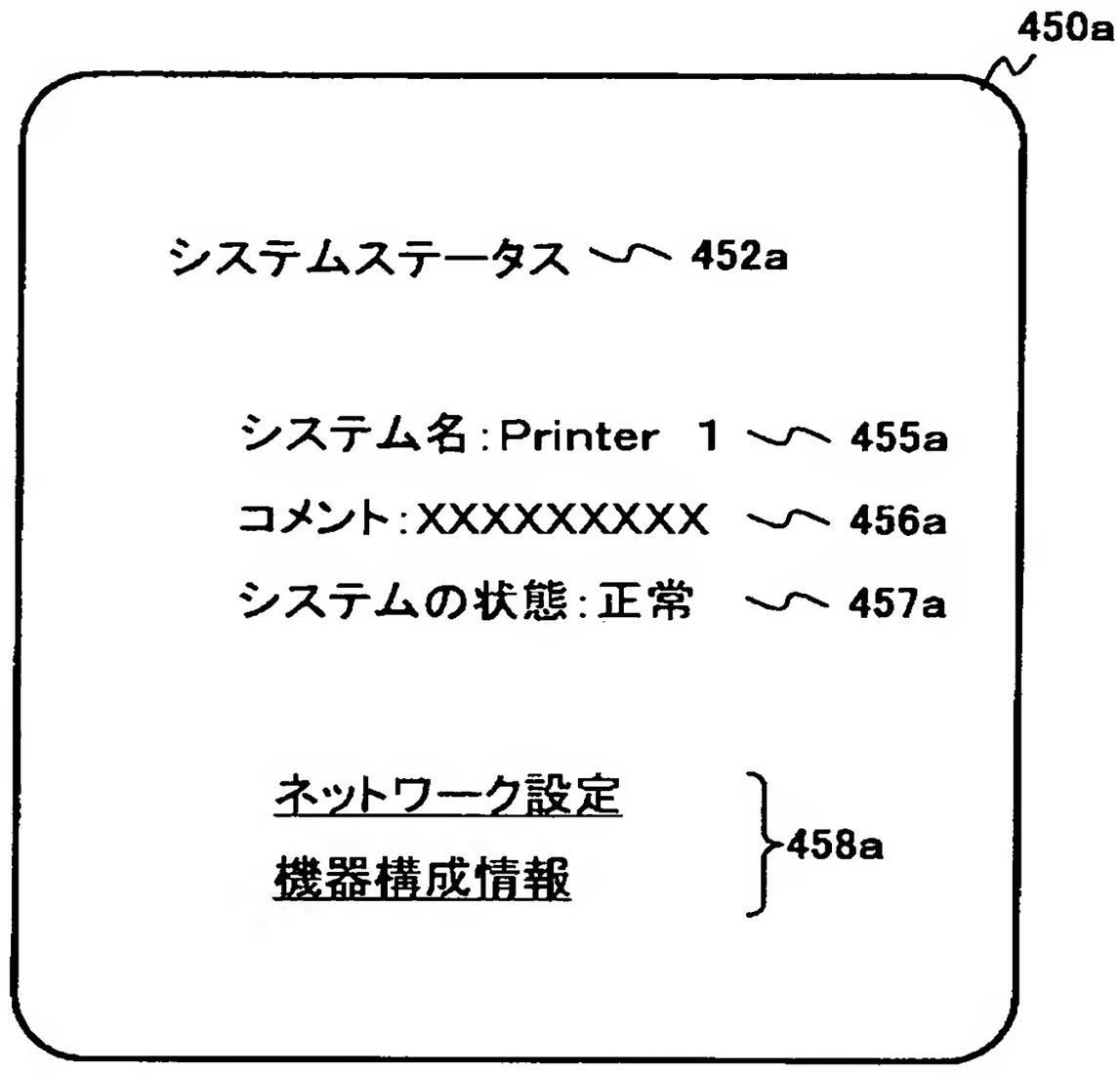
topPage.cgiの処理結果を示す
PDA端末用のHTMLの記述例を示す図

```
<html>

<head>
<meta http-equiv="Content-Language" content="ja">
<meta http-equiv="Content-Type" content="text/html; charset=shift_jis">
<meta name="GENERATOR" content="Microsoft FrontPage 4.0">
<meta name="ProgId" content="FrontPage.Editor.Document">
<title>システムステータス</title>
</head>

<body>

<p><b><font size="3">システムステータス</font>
</b></p>

<p>システム名:Printer1</p>
<p>コメント:XXXXXXXXX</p>
<p>状態:正常</p>
<p></p>
<p><a href="../webNetsettingApl/netsetting.cgi">ネットワーク設定</a></p>
<p><a href="../webSysconfigApl/sysconfig.cgi">機器構成情報</a></p>

</body>

</html>
```

440a
443a
442a
445a
446a
447a
448a
441a

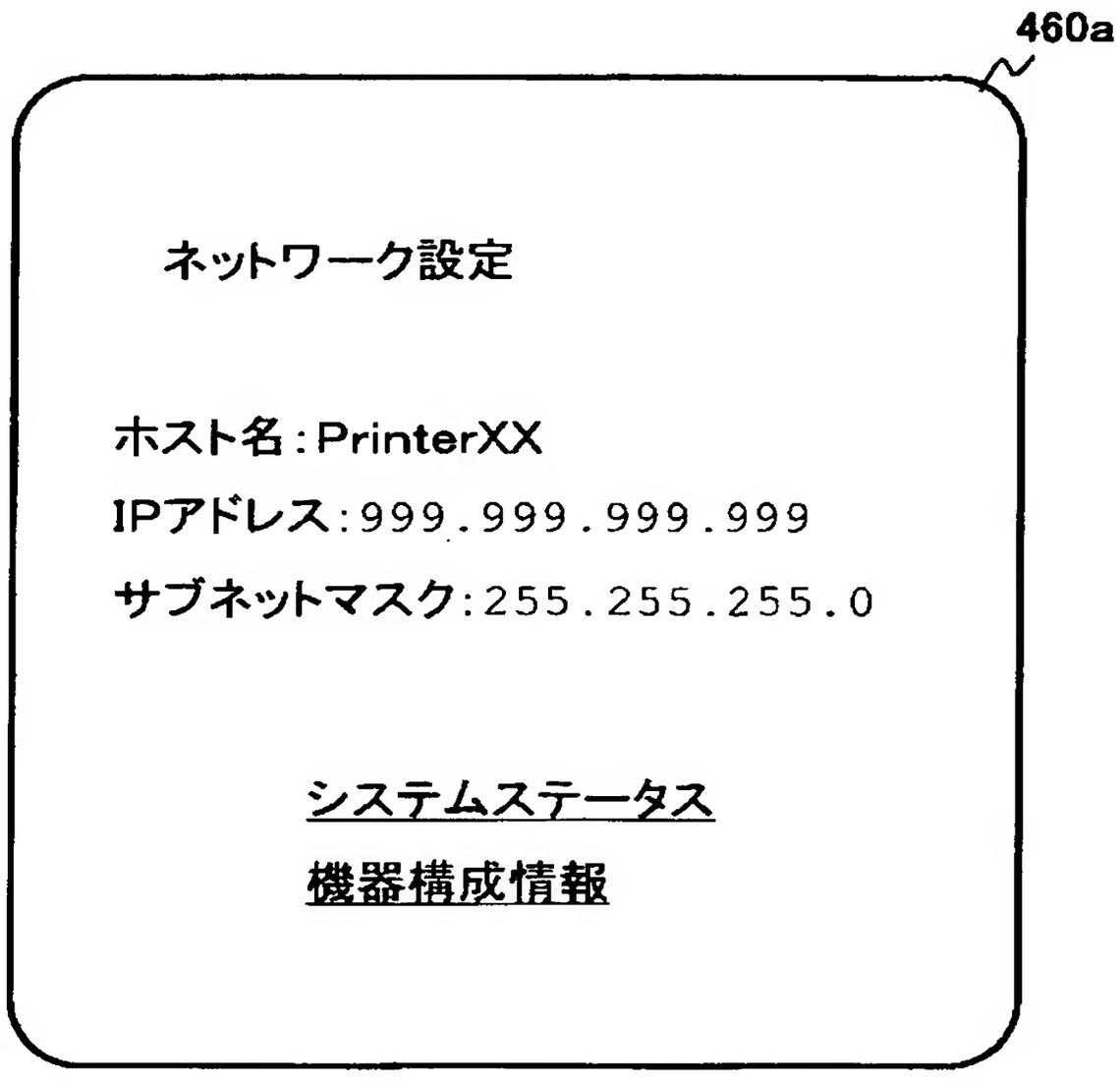
【図２５】

PDA端末でのシステムステータスの表示例を示す図



【図２６】

PDA端末でのネットワーク設定の表示例を示す図



【図２７】

一般PC用のページをPDAにそのまま表示した例を示す図

【書類名】　　要約書

【要約】

【課題】　本発明の課題は、Ｗｅｂアプリケーションを跨るページ遷移の際にもアクセスしている端末種別の継承を可能とし、ページ遷移の際のユーザの利便性を損なうことなく端末の画面表示域に応じた適切な情報提供を可能とする、複数のＷｅｂアプリケーションを有する情報処理装置を提供することを目的とする。

【解決手段】　本発明の課題は、ネットワークを介して接続される端末から第一要求を受信すると、該端末の端末種別を特定し、その特定した端末種別を示す端末種別情報を付加した該第一要求に応じたＷｅｂ情報へのパスを基準パスとして該端末側からアクセスさせる基準Ｗｅｂ情報を生成する基準Ｗｅｂ情報生成手段と、上記基準Ｗｅｂ情報を上記端末の上記第一要求に対する応答として送信することによって、上記基準パスによって該端末から上記Ｗｅｂ情報を要求する第二要求を受信するようにした情報処理装置によって達成される。

【選択図】　図６

特願２００３－１９７８５０

出　願　人　履　歴　情　報

識別番号　　［０００００６７４７］

１．変更年月日　　１９９０年　８月２４日
［変更理由］　新規登録
住　所　東京都大田区中馬込１丁目３番６号
氏　名　株式会社リコー

２．変更年月日　　２００２年　５月１７日
［変更理由］　住所変更
住　所　東京都大田区中馬込１丁目３番６号
氏　名　株式会社リコー